Operation and Maintenance

ASN-24 COMPUTER

and Associated Test Equipment

LIBRASCOPE DIVISION
GENERAL PRECISION, INC.

LIBRASCOPE / GLENDALE BRANCH   808 WESTERN AVENUE   GLENDALE, CALIFORNIA

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

Section Page

IV (Cont.)

TABLE OF CONTENTS (Cont.)

Section Page

IV (Cont.)

TABLE OF CONTENTS (Cont.)

Section Page

TABLE OF CONTENTS (Cont.)

Section Page

VII (Cont.)

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont.)

Section I
INTRODUCTION

1-1. PURPOSE OF MANUAL

This manual contains operation and maintenance instructions for the ASN-24 Computer and associated test equipment, designed and manufactured by Librascope Division, General Precision, Inc., and is written to the level of field engineers and technicians who are skilled in the handling of digital computers.

1-2. PURPOSE OF EQUIPMENT

The ASN-24 Computer is a versatile, general-purpose, electronic digital computer which, by virtue of its non-fixed internally-stored program, is easily adaptable for many commercial, scientific, and military uses. In addition, the computer's small size and weight and its low power requirements make it well suited for application in compact systems.

1-3. DESCRIPTION

The computer occupies 0. 55 cubic feet of space, weighs 31 pounds including its magnetic memory drum, and has a memory capacity of 2560 words. An input-output unit, expressly designed for the specific computer application, is used with the ASN-24.

1-4. POWER REQUIREMENTS

The computer requires the following DC power:

Voltages	Watts
+60	31.7
+35	22.3
+23	6.1
+20	.7
-8	2.0
-20	17.4

The computer also requires 30 watts of 115V AC, 400 cycles, 3 phase, assuming a power factor of 50 percent. A total of 110 watts of AC and DC power are required.

Section II
PHYSICAL DESCRIPTION

2-1. WEIGHT AND VOLUME

The computer weighs 31 pounds and is 0.55 cubic feet in volume.

2-2. MAGNETIC MEMORY DRUM

The magnetic memory drum has a 4-1/2-inch diameter, is 3-1/2 inches long, and is driven by a 400-cycle, three-phase, hysteretic synchronous motor at 6000 rpm. The drum surface has 1600 equidistant grooves along its full length. The grooves are filled with magnetic material to a depth of 0.0015-inch. Sixty-three magnetic heads are mounted 0.001 inch from the surface of the drum. Each head can magnetize the material in the grooves to a 0.050-inch wide channel or track. The heads are spaced to provide for 47 separate channels (some have more than one monitoring head). Channels are subdivided into 25-bit words, 64 words per channel. Two of the main design features of the drum are:

a. The bearings, housing, drum, and associated structure are designed to ensure a change in read-record head-to-drum spacing of less than 0.0001 inch in an airborne environment.

b. The motor is designed as an integral part of the drum, producing a lighter and more compact assembly.

2-3. READ-RECORD HEADS

a. Size: case, 0.4-in. diameter by 0.75-in. long; core, 0.3-in. long by 0.25-in. wide with 0.05-in. square cross-section.

b. Material: case, aluminum; core, ferramic H.

c. Core gap: 0.001 in. with silver shim filling gap.

d. Turns: read, 600; write, 100 center tap.

e. Shielding: 0.04-in. thick Mu metal.

f. Inductance: read, 16 millihenries; write, 0.06 millihenries for half-head.

The core is secured to the case as close to the gap as possible so that temperature variations will result in small absolute variations in drum surface-to-head spacing. The cores are mechanically mounted in the case and are entirely enclosed (except for pole pieces), thereby preventing entrance of foreign particles.

2-4. CIRCUIT CARDS

The computer has 19 circuit cards. Nine are large, 9-1/2 × 6 in., and ten are small, 4 × 6 in. (see figure 2-1). Each large one has two 37-pin connector sections; each small one has one 37-pin connector section. Listed below are each type of card and identification numbers.

a. Five large flip-flop cards, numbers 1, 2, 3, 4, and 9.

b. Two large write amplifier cards, numbers 5 and 6.

c. Two large read amplifier cards, numbers 7 and 8.

d. Four small flip-flop cards, numbers 15, 16, 17, and 19.

e. Two small read amplifier cards, numbers 11 and 12.

f. Two small head switching cards, numbers 10 and 18.

g. One small emitter follower card, number 13.

h. One small write amplifier and clock generator card, number 14.

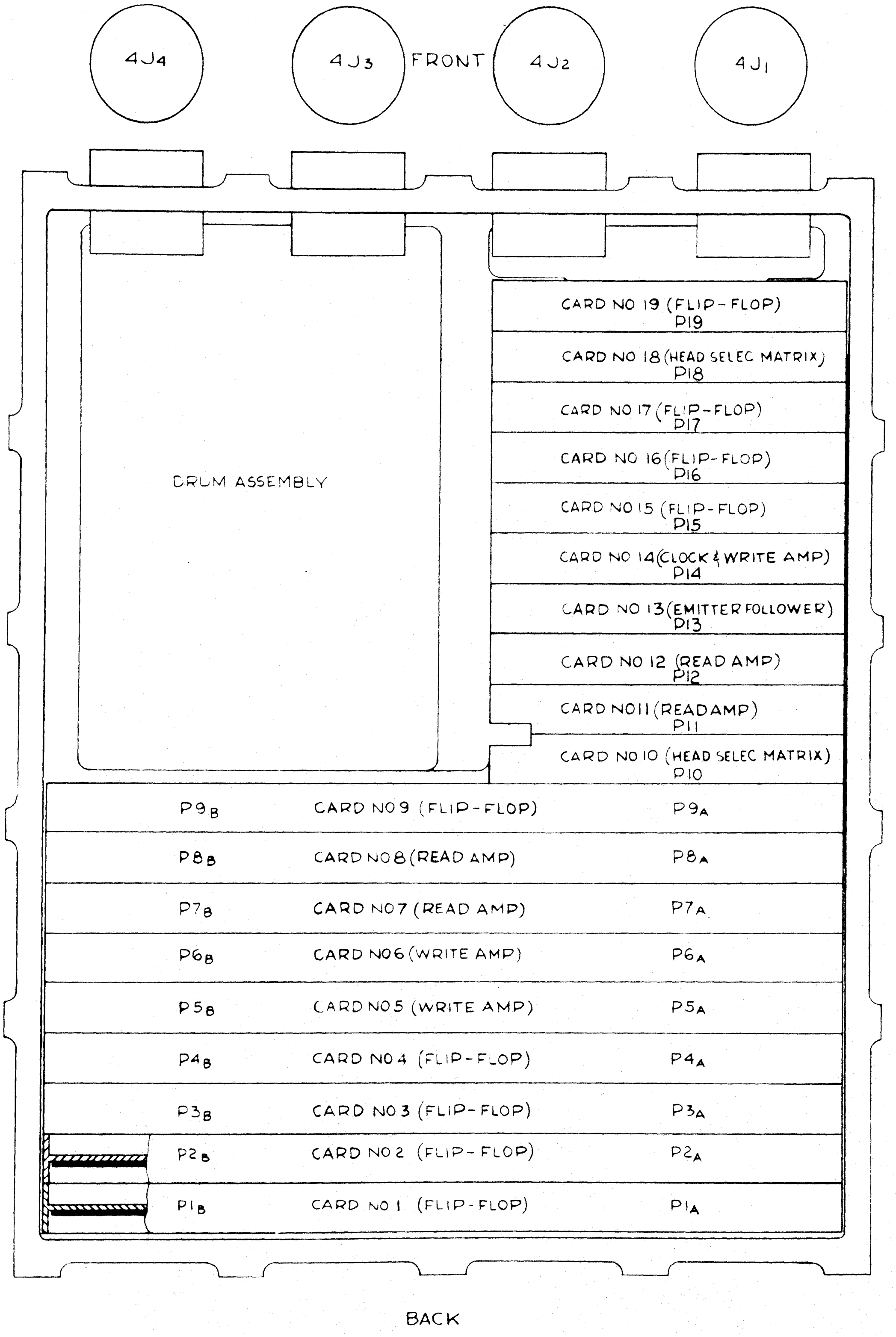

4J4
4J3
FRONT
4J2
4J1
DRUM ASSEMBLY
CARD NO 19 (FLIP-FLOP)
P19
CARD NO 18 (HEAD SELEC MATRIX)
P18
CARD NO 17 (FLIP-FLOP)
P17
CARD NO 16 (FLIP-FLOP)
P16
CARD NO 15 (FLIP-FLOP)
P15
CARD NO 14 (CLOCK & WRITE AMP)
P14
CARD NO 13 (EMITTER FOLLOWER)
P13
CARD NO 12 (READ AMP)
P12
CARD NO 11 (READAMP)
P11
CARD NO 10 (HEAD SELEC MATRIX)
P10
P9B CARD NO9 (FLIP-FLOP) P9A
P8B CARD NO8 (READ AMP) P8A
P7B CARD NO7 (READ AMP) P7A
P6B CARD NO6 (WRITE AMP) P6A
P5B CARD NO5 (WRITE AMP) P5A
P4B CARD NO4 (FLIP-FLOP) P4A
P3B CARD NO3 (FLIP-FLOP) P3A
P2B CARD NO2 (FLIP-FLOP) P2A
P1B CARD NO1 (FLIP-FLOP) P1A
BACK

Figure 2-1. Computer Component Location

Section III
ELECTRICAL DESCRIPTION

3-1. GENERAL

All electronic circuits in the computer are designed to operate under severe environmental conditions and to tolerate large differences in individual component parameters and a wide variation in supply line voltage.

3-2. COMPONENTS

3-3. Resistors

The resistors are 1/4-watt, carbon composition-type, have low dielectric loss, high DC resistivity, and high thermal shock resistance.

3-4. Semiconductors

The semiconductors in the ASN-24 Computer are silicon transistors and diodes with high back resistance and low leakage characteristics at high temperatures. The higher collector voltage ratings of silicon transistors permit larger logic swings. This reduces the susceptibility of the computer to noise disturbance.

3-5. Magnetics

The magnetic heads and flip-flop input transformers have ferrite cores. Although ferrites tend to change their magnetic characteristics over temperature extremes, the ferrites in the ASN-24 Computer have a comparatively flat permeability curve over the operating temperature range and have a minimum decrease in saturation flux density with increasing temperature. Silicon steels and permalloy, which hold their magnetic characteristics over a large temperature range, are used in some transformers.

3-6. Capacitors

Most of the capacitors are a solid tantalum type, have high dielectric and insulation strength, and have no derating of voltage over a large temperature range. The very small capacitors are the subminiature ceramic type.

3-7. CLOCK AMPLIFIER

a. Type: DC preamplifier, trigger-shaping stage, blocking oscillator, power stage.

b. Transistors: three 2N337, preamplifier; one 2N337,blocking oscillator; one 2N699, power stage.

c. Input: 300 millivolts minimum into 5000 ohms.

d. Output: 1.5 watts peak; 12-volt negative pulse and 5-volt positive pulse (read sync), each of 1 μsec duration; rise time, 0.15 μsec; and fall time, 0.4 μsec (fully loaded).

e. PRF: 100 to 200 kc.

f. Total power dissipation: 900 milliwatts.

Because a pulse amplification-clipping technique is employed in the preamplifier section, there is no change in phase shift, output relative to input, as a function of input amplitude, frequency, or temperature. Stabilization in the regenerative feedback circuit of the blocking oscillator makes the pulse width independent of temperature and transistor parameters. A DC gain of less than one provides a high degree of stability in the DC preamplifier. See figure 3-1.

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12 AND MIL-STD-103.
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15.
3. SELECT VALUE OF R12 AT FINAL TEST, NOMINAL VALUE IS 510 Ω
4. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
5. REFER TO DRAWING L 536001508 FOR ASSEMBLY DRAWING.

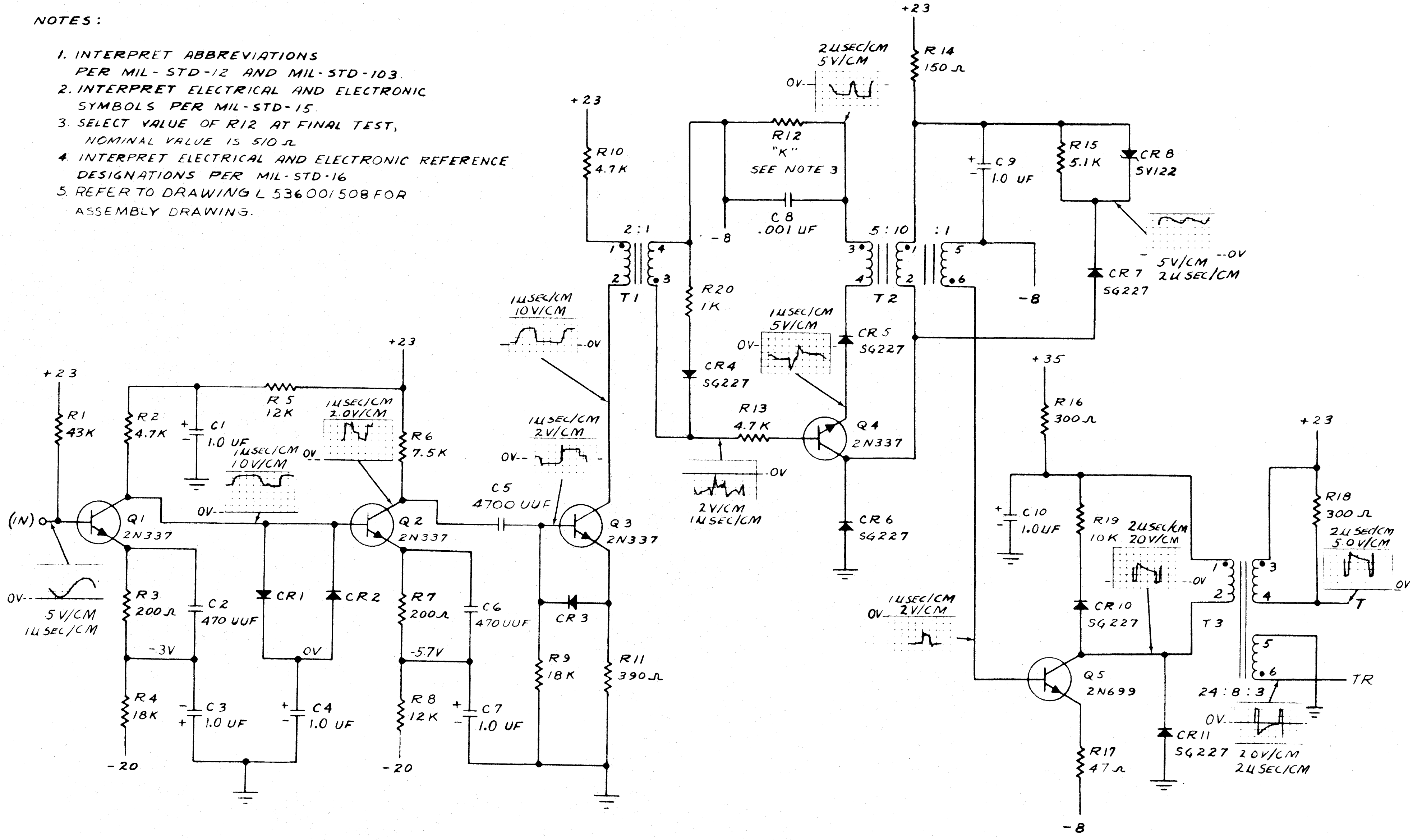


Figure 3-1. Clock Amplifier, Schematic Diagram

3-8. FLIP-FLOPS

a. Type: bi-stable Eccles-Jordan with output buffers and power gain stages. Logically a J-K type flip-flop (inherent cross-coupled feedback from the output of one side to the input of the other).

b. Transistors: two 2N337, flip-flop; two 2N337, emitter follower buffers; and two 2N699, power gain stages.

c. Input: clock pulse six volts minimum; 1 μsec duration and 5 milliwatts; transformer-coupled.

d. Output: 26 volts (+9 to +35 volts); 1 μsec rise time and 0.5 μsec fall time; drive approximately 40 gates at 0.55 ma/gate.

e. PRF: 0 to 200 kc.

f. Total power dissipation: 300 milliwatts (fully loaded).

Collector saturation in all transistors is prevented by a technique which also compensates for transistors' gain variation. Compensation is provided for change in leakage currents. The lower level of output is regulated to ensure that an equal division of load current is supplied to a multiple input AND gate. See figure 3-2.

3-9. DIODE GATES

a. Type: AND-OR pyramid with a flip-flop input at apex.

b. Diodes: HD6621.

c. Input: +9 to +35 volts.

d. Current transfer ratio: 0.8 maximum.

e. PRF: 300 kc maximum.

f. Power dissipation: 28 milliwatts per AND-OR gate.

The number of AND and OR inputs permitted in a single gate structure exceeds that required for instrumenting the logic.

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12.
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15.
3. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16.
4. REFER TO DRAWINGS L536001504, L536001505 & L536001506 FOR ASSEMBLY DRAWINGS.
5. ON CARDS NO. 9 & NO 19 (INCREMENTAL INPUT FLIP-FLOP) "T" DIODES CR5 & CR15 WILL BE CONNECTED TO +23

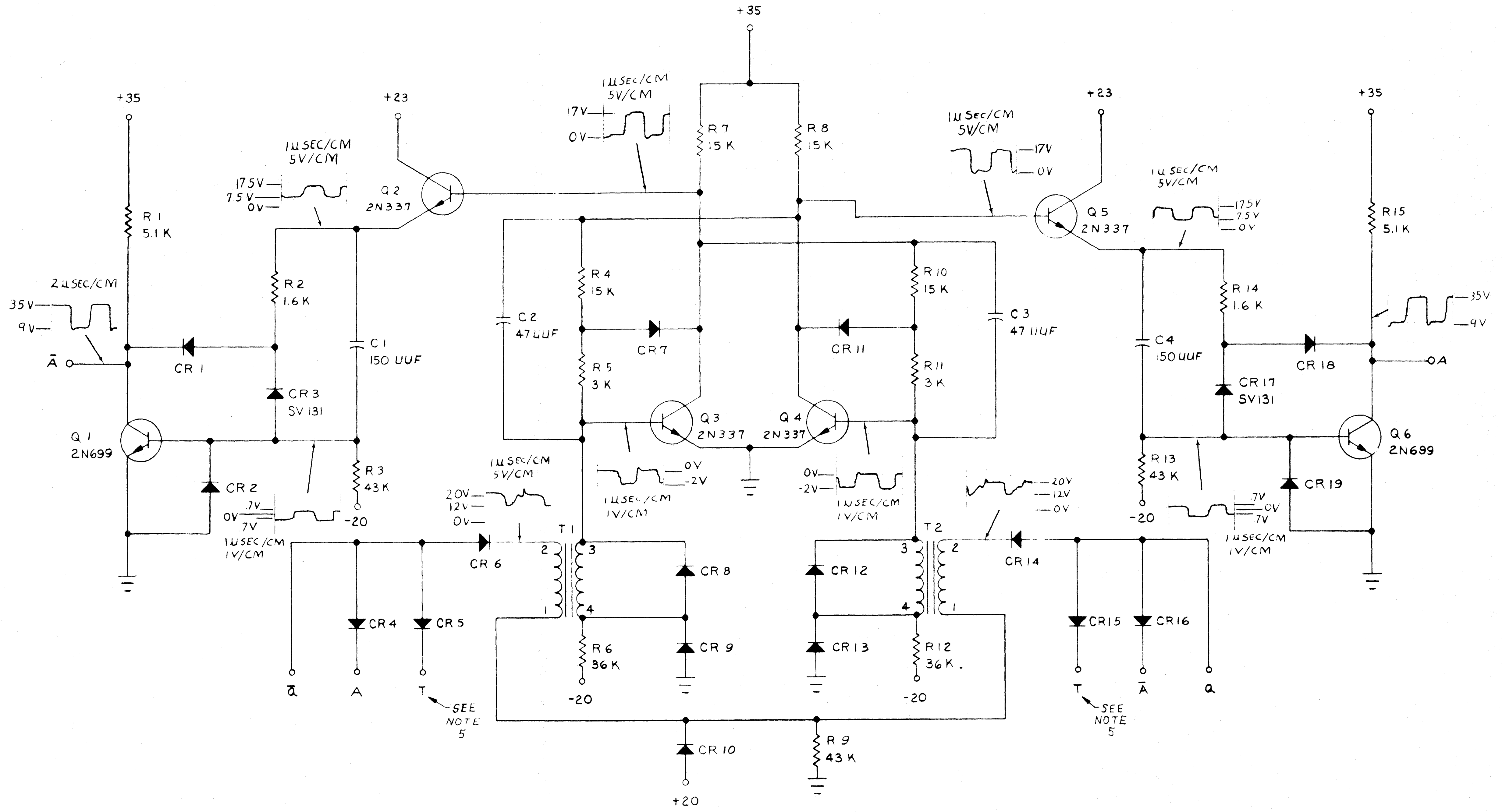


Figure 3-2. Flip-Flop, Schematic Diagram

3-10. EMITTER FOLLOWERS

a. Type: standard.

b. Transistors: one 2N697 per emitter follower.

c. Input: direct drive from the logic (+9 to +35 volts).

d. Output: direct drive from the emitter (+9 to +35 volts) with logic pull-down resistor.

e. PRF: 0 to 500 kc.

Transistors are allowed to saturate as there is no minority carrier storage problem in this configuration. This permits minimum dissipation of power in transistors, maximum utilization of logic swing, and increased reliability due to reduction of components. See figure 3-3.

3-11. WRITE AMPLIFIERS

a. Type: push-pull pulse recording on milled tooth channels. For recording "one" or "zero" there is flip-flop type input, blocking oscillator, intermediate power stage, power output stage.

b. Transistors: two 2N699, blocking oscillators; two 2N341, intermediate stages; two 2N389, output stages.

c. Input: negative clock pulse, six volts minimum, 1 μsec duration, transformer-coupled.

d. Output: 300 ma current pulse of 2.5 μsec duration into 50-turn magnetic record head represents 15 ampere turns of magnetic field strength required from write head for recording; 0.5 μsec rise time and 0.7 μsec fall time.

e. PRF: 100 to 180 kc.

f. Total power dissipation: 1 watt.

The input is logically a flip-flop input of set-reset type with additional capability of inhibiting recording of both "one" and "zero". By clamping collector and/or controlling emitter current, the transistors are not allowed to saturate. The pulse width, as determined by the blocking

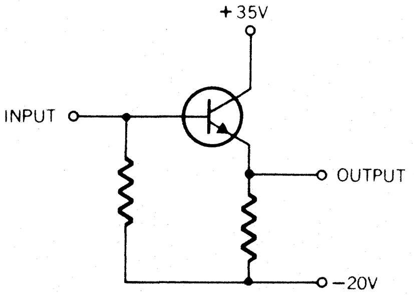


Figure 3-3. Emitter Follower, Schematic Diagram

oscillator, is independent of transistor parameters. The stability factor for transistors (dI_C/dI_{CO}) is less than two in all stages. A current-limiting feature in the output power stage prevents power runaway. See figure 3-4.

3-12. PREAMPLIFIERS

a. Type: emitter follower.

b. Transistors: one 2N495(PNP) each read head.

c. Input: 0.5 volt peak-to-peak sinusoidal into 10K load.

d. Output: 0.5 volt peak-to-peak.

e. Amplifier response: 100 to 250 kc.

f. Total power dissipation: 10 milliwatts.

The emitter follower preamplifiers are mounted close to each read head on the drum, permitting switching of heads and increased driving capabilities while reducing noise interference. See figure 3-5.

3-13. HEAD-SWITCHING MATRIX

a. Type: diode-gated.

b. Transistors: included in preamplifiers.

c. Input: Logic signals +9 to +35 volts.

d. Output: head signal (0.5 volt peak-to-peak).

e. Amplifier response: 100 to 250 kc.

f. Total power dissipation: 850 milliwatts.

3-14. READ AMPLIFIERS (READ FLIP-FLOPS)

a. Type: amplifier with modified flip-flop.

b. Transistors: two 2N337 and to 2N697, amplifier; two 2N337, gating; four 2N337 and two 2N699, flip-flop.

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12 AND MIL-STD-103
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15
3. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16.
4. SELECT VALUE OF R4 AND R5 AT FINAL TEST, NOMINAL VALUE IS 330 Ω
5. REFER TO DRAWINGS L536001508 & L536001507 FOR ASSEMBLY DRAWINGS.

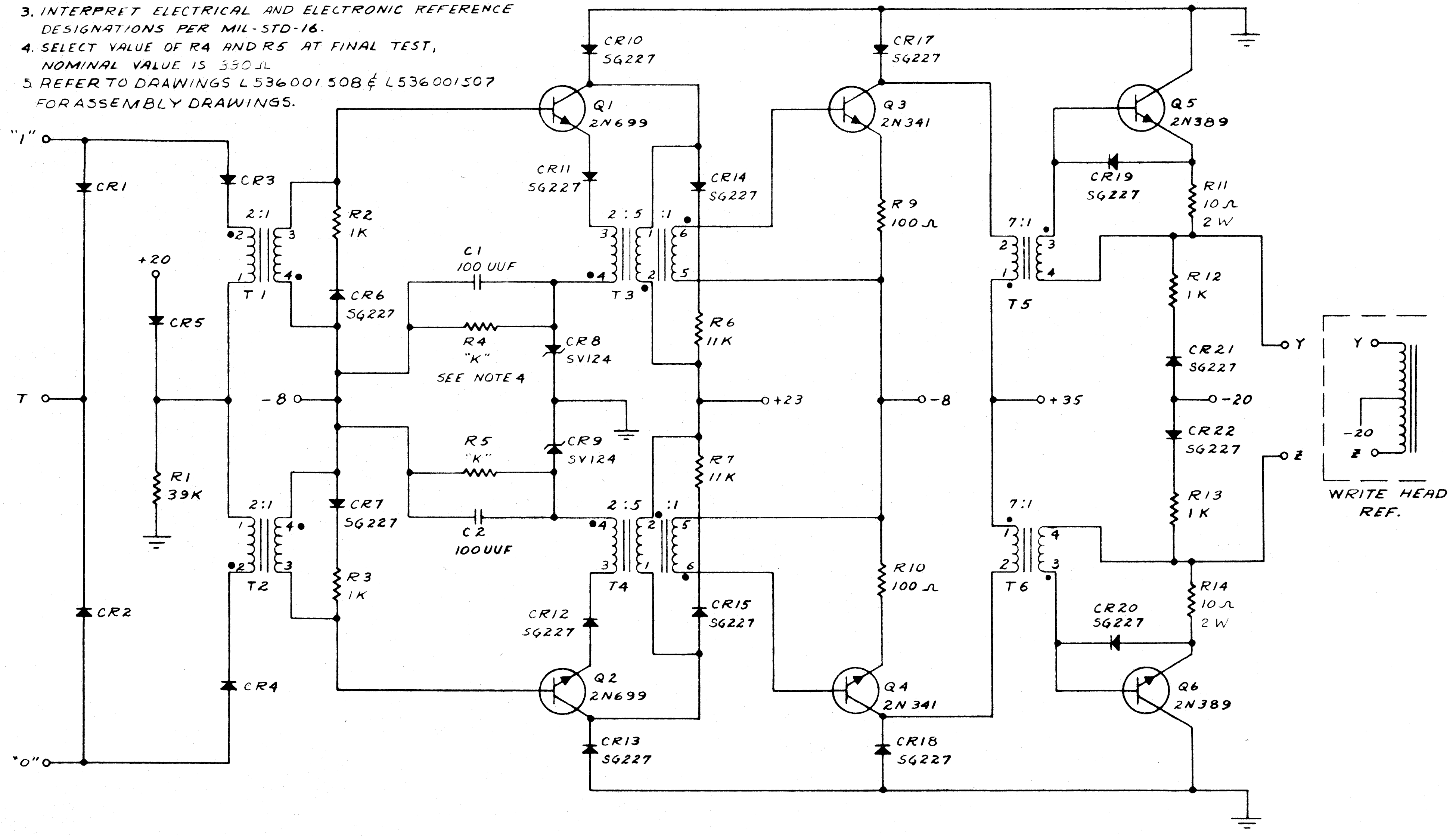
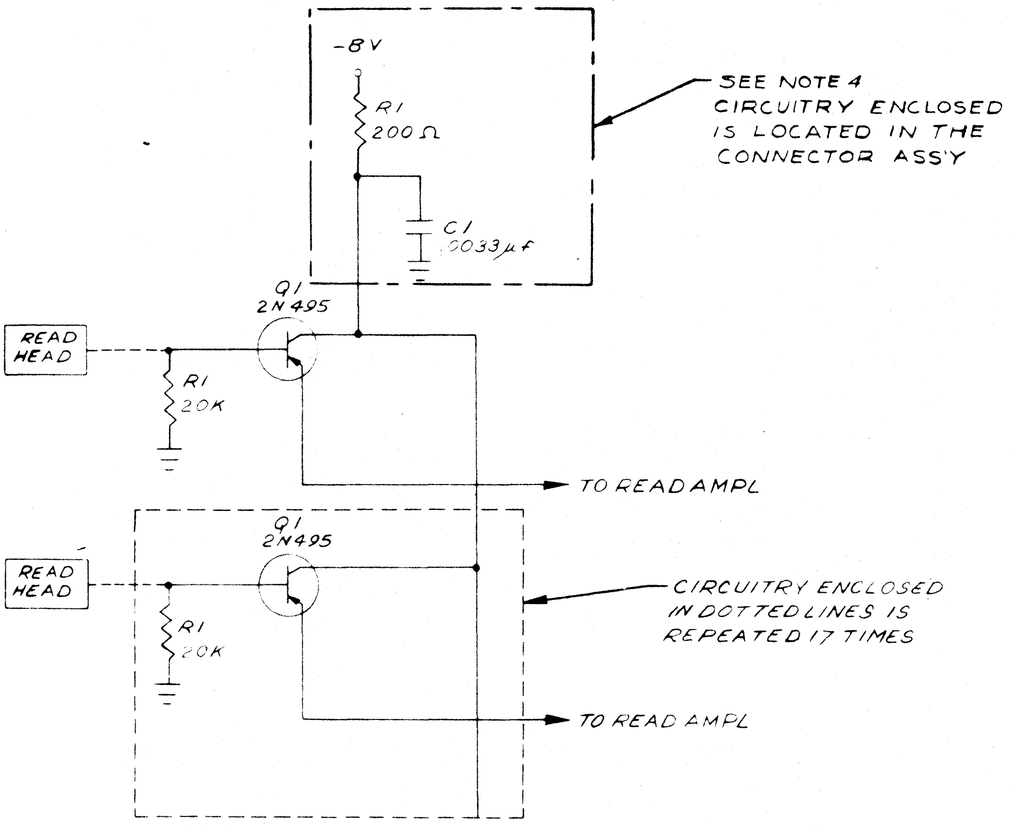


Figure 3-4. Write Amplifier, Schematic Diagram

NOTES:

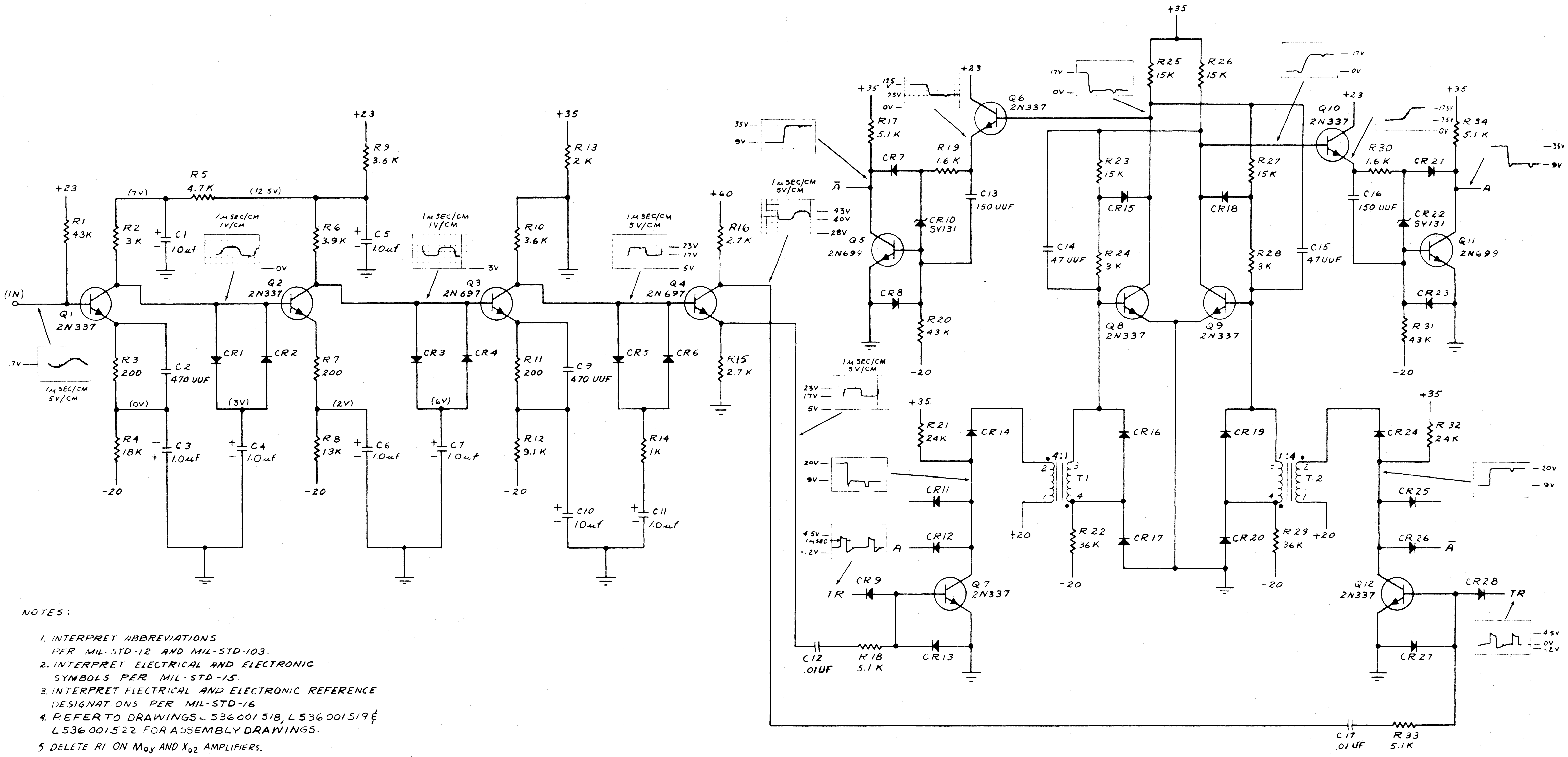
1. INTERPRET ABBREVIATIONS PER MIL-STD-12
2. INTERPRET ELECTRICAL & ELECTRONIC SYMBOLS PER MIL-STD-15.
3. INTERPRET ELECTRICAL & ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16.
4. REFERENCE DRAWINGS; L 520 002 005, CONNECTOR ASSEMBLY PRE-AMP & L 200 003 049, PREAMPLIFIER ASSEMBLY.



SEE NOTE 4 CIRCUITRY ENCLOSED IS LOCATED IN THE CONNECTOR ASS'Y

CIRCUITRY ENCLOSED IN DOTTED LINES IS REPEATED 17 TIMES

Figure 3-5. Preamplifier, Schematic Diagram



NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12 AND MIL-STD-103.
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15.
3. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
4. REFER TO DRAWINGS L 536001 518, L 536001519 & L 536001522 FOR ASSEMBLY DRAWINGS.
5. DELETE R1 ON M_{0y} AND X_{02} AMPLIFIERS.
6. X_{01} AND Y_{01} READ AMPLIFIERS TIE CR11 AND CR25 (CATHODE ENDS) TO THEIR RESPECTIVE EMITTER FOLLOWERS.

Figure 3-6. Read Amplifier, Schematic Diagram

c. Input: 300 millivolts minimum into 5000 ohms.

d. Output: standard flip-flop.

e. Amplifier response: 100 to 250 kc.

f. Total power dissipation: 850 milliwatts.

A DC gain of less than one in the amplifier provides a high degree of stability. Collector saturation is prevented in all stages. The signal is symmetrically clipped and shaped independently of changes in the operating point and parameters of the transistors. To eliminate noise and feedback through the power supply, collectors of transistors in the amplifier stages are decoupled. See figure 3-6.

3-15. DRUM

a. Type: milled; hysteretic synchronous motor, 400 cycles, 3 phase.

b. Tracks: 47.

c. Bits per track: 1600.

d. Bit rate: 160 kc at 6000 rpm.

e. Input: magnetic field from write head.

f. Output: read by read head (non-destructive read-out).

g. Total power dissipation: 30 watts AC (steady state).

Section IV
THEORY OF OPERATION

4-1. GENERAL

The ASN-24 Computer performs arithmetic and logical operations called orders. These orders include such operations as add, subtract, multiply, divide, extract, and conditional transfer. An order is a part of an instruction; other parts of an instruction define the address of the operand on which the order is to be performed and the address of the next instruction to be performed. Instructions are sequentially arranged to form a program which enables the computer to solve mathematical operations of various complexities. A program, when internally stored on the computer's memory drum, governs over-all computer operation, whereas a system of internally-fixed logic and control patterns governs the method by which the computer performs each type of instruction. It is essentially the built-in logic and control system, and not a specific program, that is the concern of this section on theory of operation.

Information stored in the computer is of two general types: control data and numerical data. Control data governs machine operation; numerical data is processed under specific control conditions. All information in the computer is represented by electrical pulses within the circuitry and by magnetic polarities on the memory drum. Two voltage levels, or magnetic polarities, are significant: a high voltage level (+35 volts) or magnetic polarization in the north direction represents the presence of a condition, or "1" state; a low voltage level (+ 9 volts) or magnetic polarization in the south direction represents the absence of a condition, or "0" state. Slight amplitude variations are meaningless. The significance of the conditions thus represented is dependent on the point and time at which they occur. Thus, a high and low voltage, respectively, may signify at various points and times any of the following: the binary digits 1 and 0 to some power of the base 2, the arithmetic signs - and +, or simply the presence or absence of some specific control condition. Information is stored magnetically over extended periods of time on an oxide-coated, rotating drum. During brief intervals, when information is being processed, it is stored in flip-flop circuits.

4-2. BASIC LOGICAL ELEMENTS

Any presentation of the theory of operation of the ASN-24 computer must employ logical equations and equivalent pictorial symbols to represent the electronic circuitry. Through use of these symbols, all unnecessary repetition of standard electronic details can be avoided. At the same time, all pertinent information can be presented concisely. The basic electronic elements are described in section III.

Although the basic electronic circuits used in the computer are relatively simple, the computer as a whole is a complex machine. This complexity arises from the fact that the computer is essentially an elaborate switching network with many switching configurations. A knowledge of configurations which define the theory of operation is necessary for successful diagnosis of machine malfunctions.

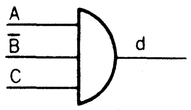
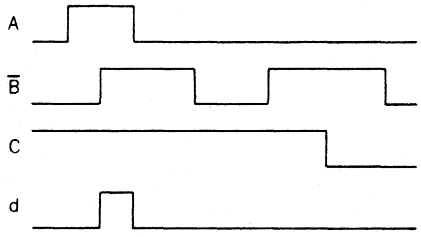
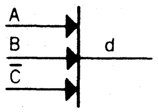
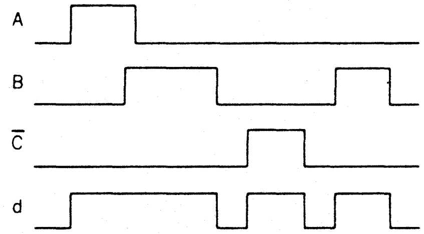
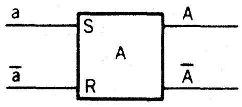
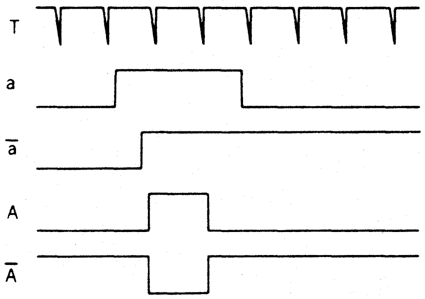
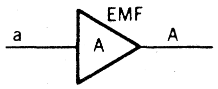
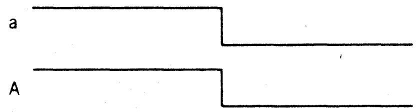
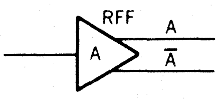
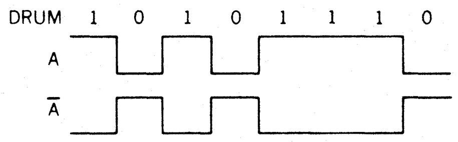
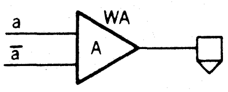
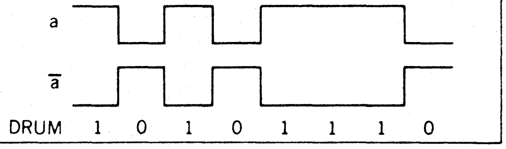
The symbology used to represent the basic elements is shown in figure 4-1 and the logical functions of these elements are described in the following paragraphs.

4-3. AND Gate

An AND gate is a logical device whose output follows its lowest input level. When all inputs are high, the output is high. When any or all inputs are low, the output is low. An AND gate is represented in row a of figure 4-1.

4-4. OR Gate

An OR gate is a logical device whose output follows its highest input level. When any or all inputs are high, the output is high. When all inputs are low, the output is low. An OR gate is represented in row b of figure 4-1.

	CIRCUIT NAME	CIRCUIT SYMBOL	CIRCUIT EQUATION	CIRCUIT WAVEFORMS
a	AND		$d=A\overline{B}C$	
b	OR		$d=A+B+\overline{C}$	
c	FLIP-FLOP		$A=a\overline{A}T$ $\overline{A}=\overline{a}AT$	
d	EMITTER FOLLOWER		$A=a$	
e	READ FLIP-FLOP		$A=``1"_{DRUM}$ $\overline{A}=``0"_{DRUM}$	
f	WRITE AMPLIFIER		$``1"_{DRUM}=a$ $``0"_{DRUM}=\overline{a}$	

NOTE:
DESIGNATIONS AND WAVEFORMS ARE ARBITRARY AND DO NOT DEFINE ANY ACTUAL CIRCUITS

Figure 4-1. Circuit Symbology

4-5. Flip-Flop

A flip-flop is a bi-stable device having complementary outputs. It is used as a temporary storage medium. The two states which a flip-flop may assume are referred to as its set and reset states. A flip-flop is in the set state (also called true or 1 state) when its non-barred output is high and its barred output is low; it is in the reset state (also called false or 0 state) when its non-barred output is low and its barred output is high. A flip-flop sets, as shown in row c of figure 4-1, when its set input (designated by a non-barred lower case letter) is high, its barred output is high, and a clock pulse T occurs. Conversely, a flip-flop resets when its reset input (designated by a barred lower case letter) is high, its non-barred output is high, and a clock pulse T occurs. A flip-flop's cross-coupling and clock triggering logic terms are not represented symbolically in either the pictorials or the logic equations because this logic is internal to a flip-flop. Because of clock pulse triggering of flip-flops, a one-bit delay is created between the time a flip-flop changes state and the time its outputs have effect upon other flip-flops.

4-6. Emitter Follower

An emitter follower is a device whose output follows its input. There is no significant time delay between input and output signal levels. Emitter followers have no inherent logical function; they are used to provide power gain for heavily loaded signals. An emitter follower is shown in row d of figure 4-1..

4-7. Read Flip-Flop

A read flip-flop (also called read amplifier) is a device which receives its input from an associated read head and generates two complementary outputs. When an associated read head senses a "1" on the drum, the read flip-flop produces a high A output and a low $\bar{A}$ output; when the read head senses a "0" on the drum, the read flip-flop produces a low A output and a high $\bar{A}$ output. A read flip-flop is shown in row e of figure 4-1.

4-8. Write Amplifier

A write amplifier is a device which receives complementary inputs from a flip-flop and generates high current pulses to an associated write head. When the write amplifier receives a high "a" input, a "1" is written on the drum; when it receives a high "$\overline{a}$" input, a "0" is written on the drum. A write amplifier is shown in row f of figure 4-1.

4-9. MEMORY SECTION

The memory section is the major information storage medium in the computer. It comprises the memory drum, its associated read/record heads and their respective read flip-flops or write amplifiers, preamplifiers, and the head selecting matrix.

4-10. Memory Drum

The memory drum is a cylindrical, motor-driven unit which is rotated at a nominal speed of 100 revolutions per second. On the periphery of the drum and running parallel to its rotational axis are 1600 magnetic oxide-filled grooves. These grooves are evenly spaced around the drum. Encasing the drum is an aluminum shroud on which read/record heads are installed. A read/record head thus defines a particular line around the drum which intersects each of the 1600 grooves during one drum revolution. These 1600 bit paths are called tracks. A track may have one or more read/record heads, depending on its function.

4-11. Read/Record Heads

The read/record heads are similar; however, the function of each head is determined by its wiring to either a read flip-flop or to a write amplifier. Each read/record head is wired either to one or the other of these circuits, never to both. A read/record head wired to a write amplifier continuously records information onto its associated track; a read/record head wired to a read flip-flop continuously reads information from its associated track. A read/record head wired to a read flip-flop will subsequently be referred to as a read head; a read/record head wired to a write amplifier will be referred to as a write head. The process of writing information erases automatically any previously recorded information. No special erase process is necessary.

4-12. Tracks

The drum shroud contains 63 read/record heads which define 47 tracks on the memory drum. These tracks and their associated heads and read flip-flops or write amplifiers are shown in figure 4-2 and are defined in the following paragraphs.

4-13. Clock Track

The clock track is prerecorded with each of its 1600 grooves polarized in the same direction. The clock track has one associated read head which feeds amplification and wave-shaping circuits located in the clock amplifier (see figure 4-2). The clock amplifier generates spiked pulses at a frequency of approximately 160 kilocycles (1600 pulses per drum revolution at 100 revolutions per second). These spiked pulses, called clock pulses, provide a 12-volt negative-going pulse, T, which is used to trigger all flip-flops in the computer with the exception of the read flip-flops which are triggered by a 5-volt positive-going pulse, T. The clock pulses are used to synchronize the computer circuitry with the memory drum. This synchronization is accomplished by triggering all flip-flops with clock pulses, thereby preventing change of state of flip-flops during the intervals between clock pulses. Since the computer circuitry is synchronized from the memory drum, slight variations in the rotational speed of the drum cannot adversely affect computer operation. Setting and resetting of flip-flops is accomplished only at clock pulse time, as shown in row c of figure 4-1.

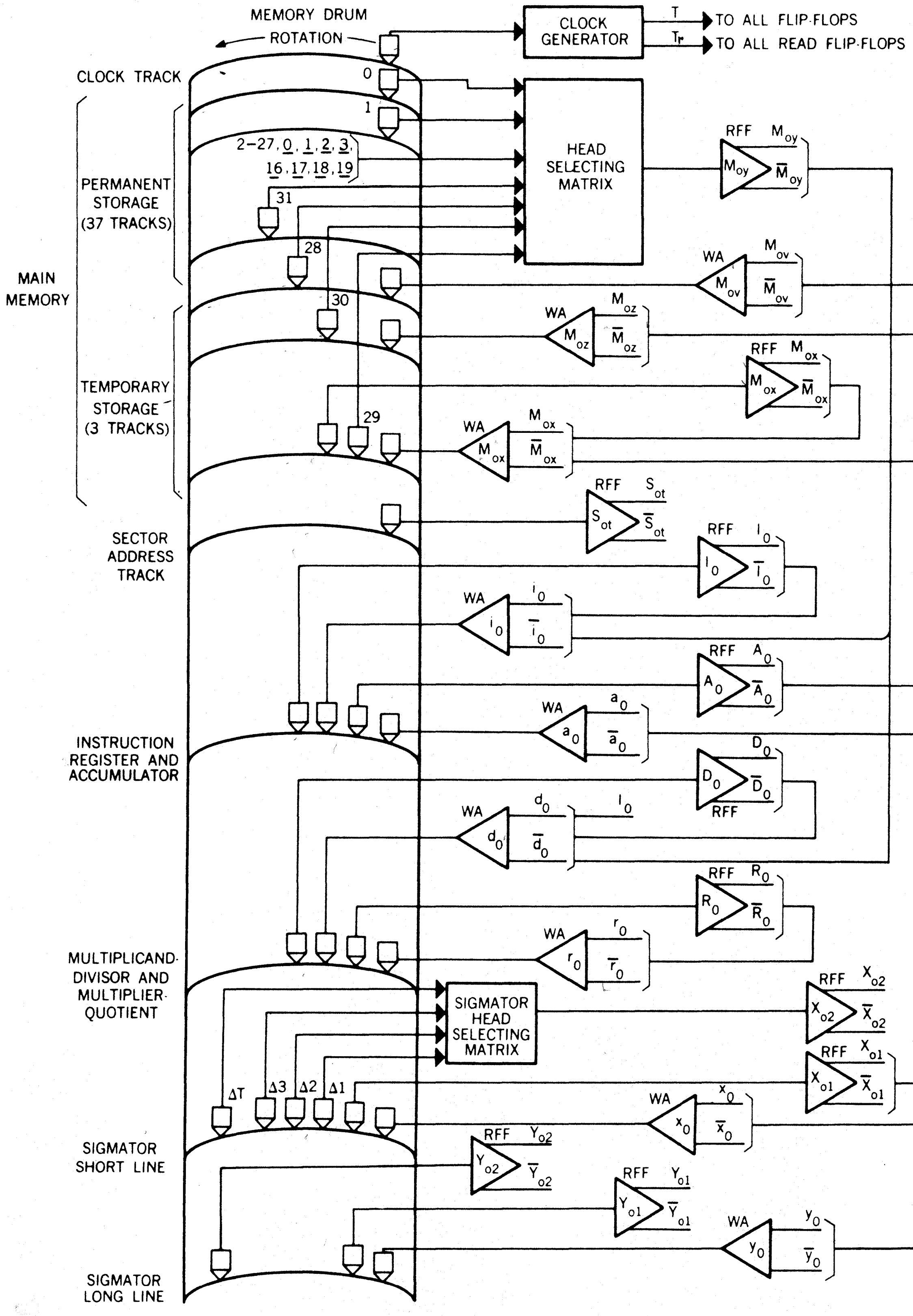

MEMORY DRUM
ROTATION
CLOCK GENERATOR
T
T_r
TO ALL FLIP-FLOPS
TO ALL READ FLIP-FLOPS
CLOCK TRACK
0
1
2–27, 0, 1, 2, 3,
16, 17, 18, 19
31
28
30
29
PERMANENT STORAGE (37 TRACKS)
MAIN MEMORY
TEMPORARY STORAGE (3 TRACKS)
HEAD SELECTING MATRIX
RFF M_{oy} $\overline{M}_{oy}$
WA M_{ov} $\overline{M}_{ov}$
WA M_{oz} $\overline{M}_{oz}$
RFF M_{ox} $\overline{M}_{ox}$
WA M_{ox} $\overline{M}_{ox}$
RFF S_{ot} $\overline{S}_{ot}$
SECTOR ADDRESS TRACK
RFF I_0 $\overline{I}_0$
WA i_0 $\overline{i}_0$
RFF A_0 $\overline{A}_0$
WA a_0 $\overline{a}_0$
INSTRUCTION REGISTER AND ACCUMULATOR
RFF D_0 $\overline{D}_0$
WA d_0 $\overline{d}_0$
I_0
RFF R_0 $\overline{R}_0$
WA r_0 $\overline{r}_0$
MULTIPLICAND-DIVISOR AND MULTIPLIER-QUOTIENT
SIGMATOR HEAD SELECTING MATRIX
RFF X_{o2} $\overline{X}_{o2}$
RFF X_{o1} $\overline{X}_{o1}$
ΔT
Δ3
Δ2
Δ1
WA x_0 $\overline{x}_0$
SIGMATOR SHORT LINE
RFF Y_{o2} $\overline{Y}_{o2}$
RFF Y_{o1} $\overline{Y}_{o1}$
WA y_0 $\overline{y}_0$
SIGMATOR LONG LINE

Figure 4-2. Memory Section

4-14. Sector Address Track

The sector address track contains permanently-recorded control information which is continuously read by the S_{ot} read flip-flop via its associated read head (see figure 4-2). The primary function of the sector address track is to define each of the 64 sectors of the main memory tracks to enable location of particular words for computing purposes. A sector is a 25-bit, one-word, storage segment. The sector address track also provides synchronization pulses which are used to initially enter, and then maintain, the pulse time indicator in synchronism with the drum. In addition, the sector address track furnishes control codes used for setting the input-output selection register and the sigmator program.

The S_{ot} signals occur at the following pulse times of each word: sector address codes occur at P_{14} through P_9 time ($s\beta$) and are repeated at P_8 through P_3 time ($s\alpha$); a "1" occurs at every P_{15} time for synchronization of the pulse time indicator; the input-output selection register control codes occur at P_{24} through P_{20} time; and sigmator program information occurs at P_2 and P_1 time. For detailed explanation on locating the sector address of a word, refer to paragraphs 4-30 and 4-33; for pulse time indicator synchronization, refer to paragraph 4-27; for input-output selection register control, refer to paragraph 4-31; and for sigmator program control, refer to paragraph 4-69.

4-15. Main Memory

Main memory comprises 40 tracks on the memory drum — 37 permanent storage tracks and 3 temporary storage tracks. Associated with main memory is the track selecting matrix and the time-shared, main memory read flip-flop, M_{oy}. The permanent storage tracks are described in paragraph 4-16, the temporary storage tracks are described in paragraph 4-17, and the head selecting matrix is described in paragraph 4-18.

4-16. Permanent Storage Tracks. The permanent storage tracks comprise 37 of the 40 main memory tracks (see figure 4-2). These 37 tracks store the program which consists of both instruction and operand words.

Each track has one associated read head numbered 0 through 27, 31, $\underline{0}$, $\underline{1}$, $\underline{2}$, $\underline{3}$, $\underline{16}$, $\underline{17}$, $\underline{18}$, and $\underline{19}$. The outputs of all 37 read heads are routed to the head switching matrix which, under control signals from the track selection register, selects the track of a desired word, then passes the word to the commonly shared main memory read flip-flop, M_{oy}. The output of M_{oy} may be routed to the instruction register, the order register, the multiplicand-divisor register, and to the serial adder via flip-flop D_1. These registers accept the M_{oy} information under specified control conditions. Normally, instruction words are routed to the instruction register. Operand words are routed to the multiplicand-divisor register and the serial adder.

4-17. Temporary Storage Tracks. The temporary storage tracks comprise 3 of the 40 main memory tracks (see figure 4-2). These three tracks store data temporarily while it is still required in the computation. Normally, the data consists of intermediate solutions such as a sum or a product which is needed for larger scale problem solutions defined by the program, or to temporarily store input-output data. The three tracks have one read head each, numbered 28, 29, and 30 which are wired to the head switching matrix. Upon selection of a read head the respective signal passes to the M_{oy} read flip-flop. Each track also has a write head and associated write amplifier. The write amplifiers, m_{ov}, m_{ox}, and m_{oz}, receive information from the accumulator register under specified control conditions. The spacing of the write heads with relation to the read heads is such that a word written on $m_{\overset{X}{oz}}$ is available at M_{oy} four word-times later, a word written on m_{oz} is available at M_{oy} 32 word times later, and a word written on m_{ov} is available at M_{oy} 60 word times later.

Track M_{ox} contains a second read head and associated read flip-flop. The output of the read flip-flop may be routed to the m_{ox} write amplifier to permit recirculation of the information, if required. The output of the M_{ox} read flip-flop is also routed to the error determining circuits where it is used for output operations, refer to paragraph 4-43. Separation between the read and write heads is equivalent to eight word times. This recirculation line, or loop, permits more rapid access to desired words contained on the M_{ox} track.

4-18. Head Selecting Matrix. The head selecting matrix is used to select any one of the 40 main memory tracks for read-out via read flip-flop M_{oy}. The head selecting matrix comprises 40 gating circuits, one for each main memory track. See figures B-16 and B-17. Each AND gate of the matrix has a unique gating code derived from various configurations of the track selection register flip-flops (S_1, S_2, S_3, S_4, S_5) and the externally located relay S_6. Refer to paragraph 4-30 for particulars on the track selection register.

When a specific code is generated by the track selection register and the S_6 relay, it is decoded by its associated gate in the matrix to electrically connect the desired main memory read head via a preamplification stage to read flip-flop M_{oy}. It may be noted that there is always an M_{oy} output i.e., M_{oy} is always copying the contents of some main memory track. However, the computer circuits only accept M_{oy} information at specific control times.

4.19 Instruction Register

The instruction register is a 25-bit recirculating register on the drum used for storing instruction words, see figure 4-2. The instruction register contains one write head and associated write amplifier i_o, and one read head and associated read flip-flop I_o. The write and read heads are spaced to provide a 25 bit delay to make the instruction word in the register available every word time.

An instruction word is routed into the instruction register from main memory via the M_{oy} read flip-flop during the instruction read-in phase. At this time recirculation of the register is blocked. During all other phases, except last word phase, the instruction recirculates.

4.20 Accumulator Register

The accumulator register is a 24-bit recirculating register used for storing operand words during computation, see figure 4-2. The accumulator register contains one write head and one read head and their respective write amplifier a_o and read flip-flop A_o. Between the read flip-flop A_o and write amplifier a_o is a serial adder which, in addition to other control logic, controls information flow into the accumulator register. The serial adder is discussed in paragraph 4-40.

The accumulator register normally receives operand words from main memory read flip-flop M_{oy} via the serial adder. While the M_{oy} operand is passing through the adder to the register, the operand stored in the register may be added or subtracted from it thus entering the sum or difference into the accumulator register. Other types of arithmetic operation may also be performed. At the time that the M_{oy} operand is passing through the adder, recirculation of the accumulator may also be blocked to permit the accumulator register to simply copy M_{oy}. The contents of the accumulator may only be altered during first word phase FW, additional words phase AW, or last word phase LW phase; during all other phases the contents of the accumulator is permitted to recirculate.

The operand word in the accumulator register may be routed to any of the three working storage registers M_{ox}, M_{oy}, or M_{oz}, or to Σ long line Y_o, in addition to being recirculated in the accumulator.

4-21. Multiplicand-Divisor Register

The multiplicand-divisor register is a 24-bit recirculating register used for storing the multiplicand during a multiplication order or the divisor during a divide order, see figure 4-2. The multiplicand-divisor register contains one write and one read head and their resepctive write amplifier d_o and read flip-flop D_o.

For detailed descriptions of how the multiplicand-divisor register is used refer to multiplication order, paragraph 4-67; divide order, paragraph 4-68; and extract order, paragraph 4-63.

4-22. Multiplier-Quotient Register

The multiplier-quotient register is a 24-bit recirculating register used for storing the multiplier during a multiplication order or the quotient during a division order, see figure 4-2. The multiplier-quotient register contains one write and one read head and their respective write amplifier r_0 and read flip-flop R_0.

For detailed descriptions of how the multiplier-quotient register is used refer to multiply order, paragraph 4-67; and divide order, paragraph 4-68.

4-23. Sigmator Long Line

The sigmator long line is used as an integrand register in the sigmator section of the computer. The sigmator long line contains one write head and associated write amplifier y_0; and two read heads and their associated read flip-flops Y_{o1} and Y_{02}. Head spacing provides an 8-word delay between input y_0 and output Y_{o1}, and a 32-word delay between input y_0 and output Y_{o2}.

Information may be received by input y_0 from the accumulator register A_0 during FW of a normal store order ($0_3\ 0_2\ 0_1\ \bar{S}_5\ \bar{S}_4\ \bar{S}_3\ \bar{S}_2\ \bar{S}_1$) or from the output of the sigmator two-input adder. Outputs Y_{o1} or Y_{o2} may be transferred to the accumulator register during modified store orders $0_3\ 0_2\ 0_1\ \bar{S}_5\ S_4\ \bar{S}_3\ S_2\ S_1$ and $0_3\ 0_2\ 0_1\ \bar{S}_5\ S_4\ \bar{S}_3\ \bar{S}_2\ S_1$, respectively. Output Y_{o2} is also sent to the error determining circuits during output operations, refer to paragraph 4-43.

4-24. Sigmator Short Line

The sigmator short line is used as an increment accumulation register in the sigmator section of the computer. The sigmator short line contains one write head and associated write amplifier x_0, and five read heads to provide increased availability of information. One read head is associated with read flip-flop X_{o1} and provide a word delay between x_0 and X_{o1}. The other four read heads ($\Delta 1$, $\Delta 2$, $\Delta 3$, and ΔT) time share read flip-flop X_{o2}. Head selection is accomplished by a head selecting matrix which is program controlled via flip-flops W_1 and W_2 of the input-output selection register, refer to paragraph 4-31. Head spacing provides the following delays with respect to X_{o1}: head $\Delta 1$ - 6 words, head $\Delta 2$ - 12 words plus 19 bits, head $\Delta 3$ - 16 words plus 13 bits, head ΔT - 29 words plus 5 bits.

Incremental information is received by input x_0 from the output of the sigmator one-increment adder-subtractor. Output X_{o1} is used as an input to the one-increment adder-subtracter, and output X_{02} is used as an input via the sigmator two-input adder to the sigmator long line.

4-25. CONTROL

The major over-all control governing computer operation is the stored program on the memory drum. Important subsidiary controls which enable the computer to follow the program include (1) the pulse time indicator, (2) the phase sequencing control, (3) the order register, (4) the track selection register, and (5) the input-output selection register.

The pulse time indicator enables designation of each pulse time of a computer word to afford precise timing of all operations. The phase sequencing control provides over-all sequential control of operations. During a particular phase, for example, only certain types of computer operation are permitted. When these operations are completed, another phase is entered thereby allowing other types of operation to be performed. The various phases normally follow one another in a limited number of fixed sequential patterns. The order, track selection, and input-output selection registers receive coded program information from the drum during specified control phases. The registers store the codes until phase control permits the codes to be sampled and acted upon.

The order register stores the order part of an instruction word, the track selection register normally stores the code of a desired track in main memory, and the input-output selection register normally stores the code of a desired converter in the input-output equipment.

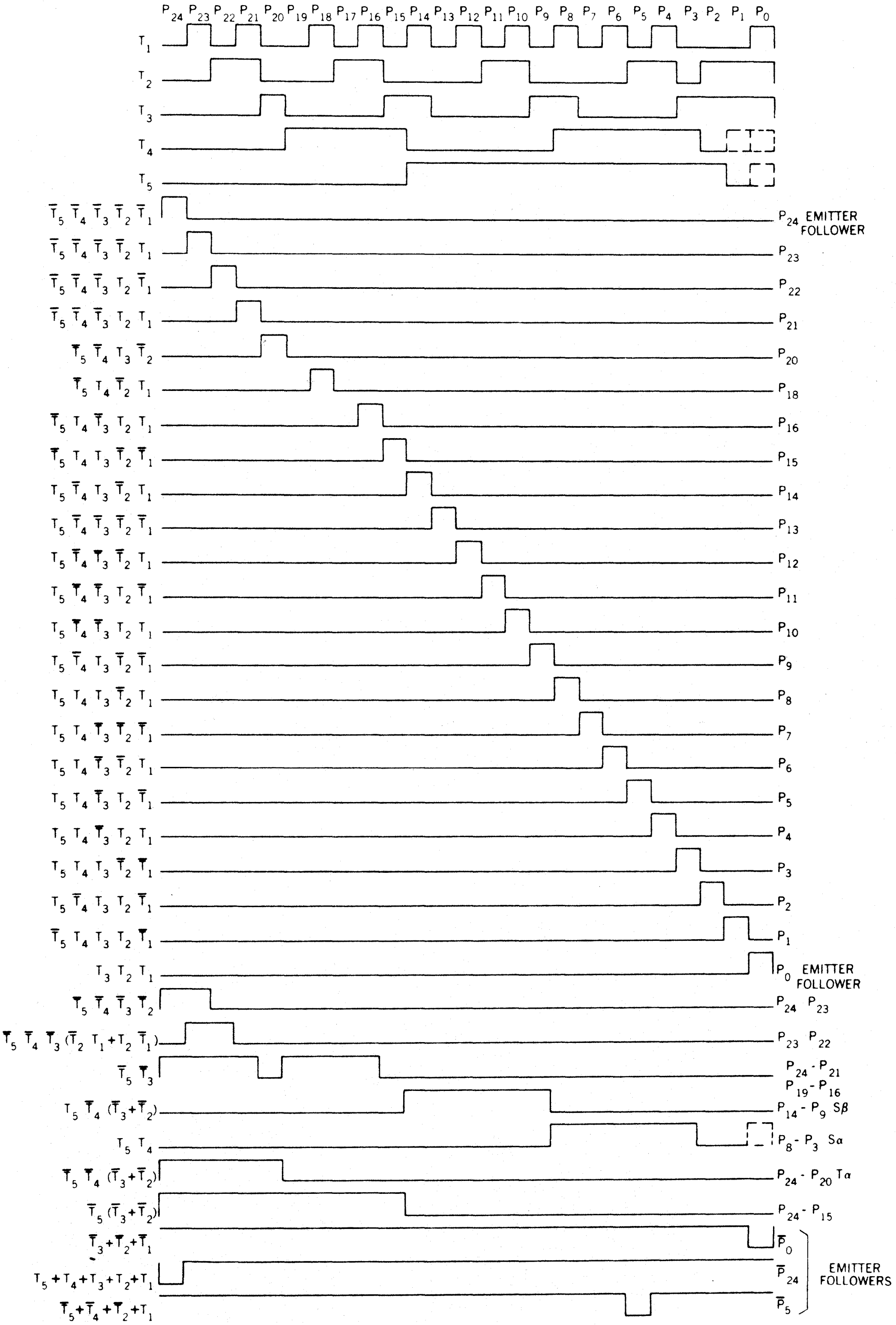
The following paragraphs provide detailed descriptions of these controls which are essential to the general purpose section of the computer. Controls peculiar to the sigmator section are separately discussed in paragraph 4-69.

4-26. Pulse Time Indicator

The pulse time indicator consists of five flip-flops and their interconnecting logic. These flip-flops, T_1 through T_5, are used to define each bit of a "computer word". The configuration of the pulse time indicator changes at each clock time. One computer word is composed of 25 bits which are termed P_{24} through P_0. The P_{24} bit occurs first, the P_0 bit last. Each of the pulse time indicator flip-flops has a distinct output. The output waveforms of these flip-flops and logical derivations of these outputs with respect to their pulse times are illustrated in figure 4-3. The pulse time indicator normally recycles every word time. The pulse time indicator logical diagram, figure 4-4 illustrates the logic for the pulse time indicator together with its logical equations.

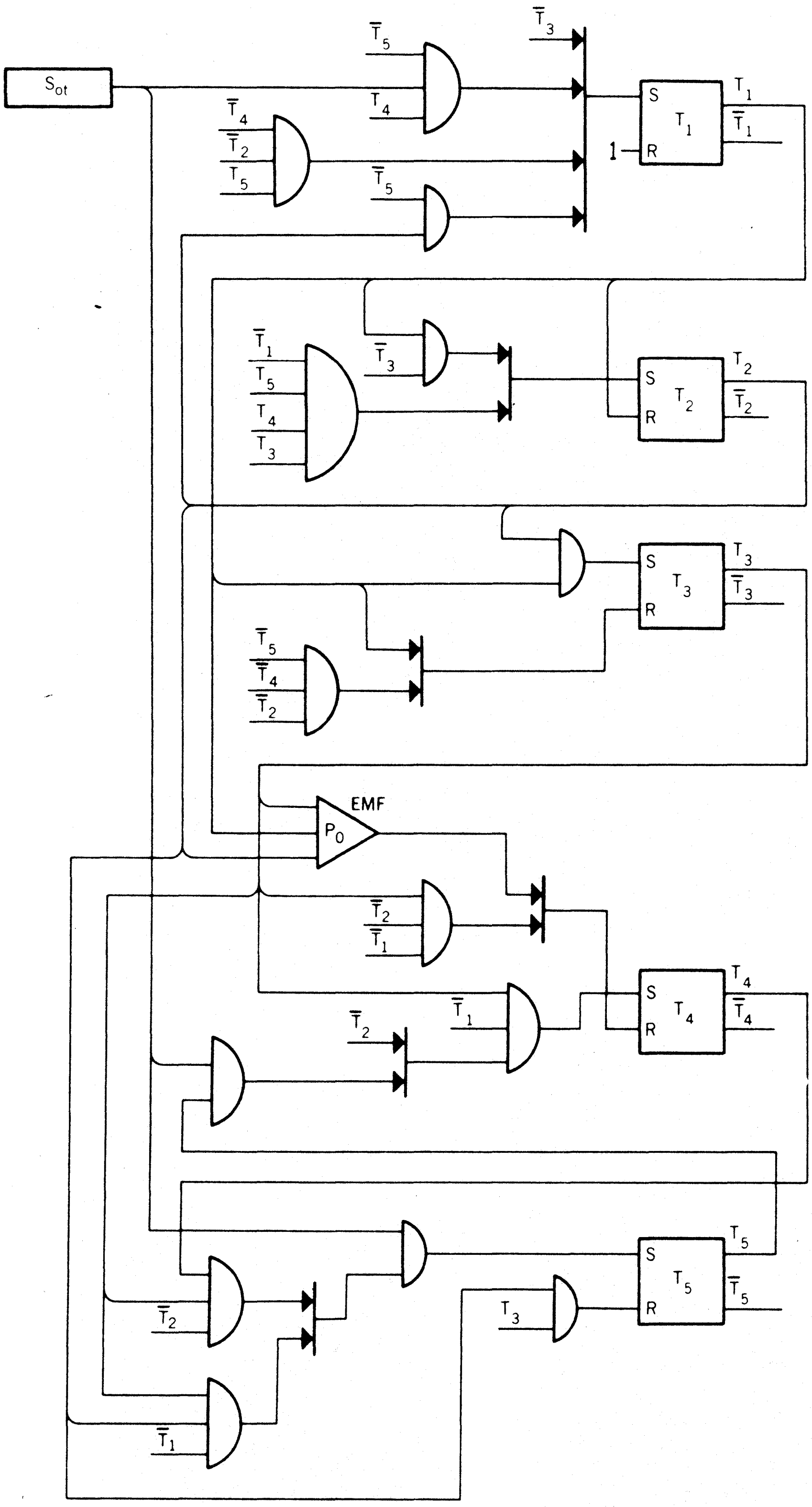
4-27. Initial Synchronization. Synchronization of the pulse time indicator is accomplished in less than two drum revolutions at computer start. This initial synchronization is accomplished by a "1" that is contained in all 64 words on the sector address track (S_{ot}) at pulse time 15 only. When the T flip-flops come on, they may or may not be defining the correct time. That is, the pulse time defined by the indicator may not be the same as the true pulse time at any read flip-flop. Whether or not the indicator is defining the correct pulse time, it will count down in a normal sequence until the P_{15} pulse is defined. At this time the indicator will sample S_{ot}. The condition of S_{ot} determines whether the indicator continues in a normal manner, or precesses to the P_{20} state. This operation is described in the following paragraphs.

At P_{15} time the pulse time indicator flip-flops assume the following states.



P_{24} P_{23} P_{22} P_{21} P_{20} P_{19} P_{18} P_{17} P_{16} P_{15} P_{14} P_{13} P_{12} P_{11} P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0
T_1
T_2
T_3
T_4
T_5
$\bar{T}_5\ \bar{T}_4\ \bar{T}_3\ \bar{T}_2\ \bar{T}_1$
P_{24} EMITTER FOLLOWER
$\bar{T}_5\ \bar{T}_4\ \bar{T}_3\ \bar{T}_2\ T_1$
P_{23}
$\bar{T}_5\ \bar{T}_4\ \bar{T}_3\ T_2\ \bar{T}_1$
P_{22}
$\bar{T}_5\ \bar{T}_4\ \bar{T}_3\ T_2\ T_1$
P_{21}
$\bar{T}_5\ \bar{T}_4\ T_3\ \bar{T}_2$
P_{20}
$\bar{T}_5\ T_4\ \bar{T}_2\ T_1$
P_{18}
$\bar{T}_5\ T_4\ \bar{T}_3\ T_2\ T_1$
P_{16}
$\bar{T}_5\ T_4\ T_3\ \bar{T}_2\ \bar{T}_1$
P_{15}
$T_5\ \bar{T}_4\ T_3\ \bar{T}_2\ T_1$
P_{14}
$T_5\ \bar{T}_4\ \bar{T}_3\ \bar{T}_2\ \bar{T}_1$
P_{13}
$T_5\ \bar{T}_4\ \bar{T}_3\ \bar{T}_2\ T_1$
P_{12}
$T_5\ \bar{T}_4\ \bar{T}_3\ T_2\ \bar{T}_1$
P_{11}
$T_5\ \bar{T}_4\ \bar{T}_3\ T_2\ T_1$
P_{10}
$T_5\ \bar{T}_4\ T_3\ \bar{T}_2\ \bar{T}_1$
P_9
$T_5\ T_4\ T_3\ \bar{T}_2\ T_1$
P_8
$T_5\ T_4\ \bar{T}_3\ \bar{T}_2\ \bar{T}_1$
P_7
$T_5\ T_4\ \bar{T}_3\ \bar{T}_2\ T_1$
P_6
$T_5\ T_4\ \bar{T}_3\ T_2\ \bar{T}_1$
P_5
$T_5\ T_4\ \bar{T}_3\ T_2\ T_1$
P_4
$T_5\ T_4\ T_3\ \bar{T}_2\ \bar{T}_1$
P_3
$T_5\ \bar{T}_4\ T_3\ T_2\ \bar{T}_1$
P_2
$\bar{T}_5\ T_4\ T_3\ T_2\ \bar{T}_1$
P_1
$T_3\ T_2\ T_1$
P_0 EMITTER FOLLOWER
$\bar{T}_5\ \bar{T}_4\ \bar{T}_3\ \bar{T}_2$
$P_{24}\ P_{23}$
$\bar{T}_5\ \bar{T}_4\ \bar{T}_3\ (\bar{T}_2\ T_1+T_2\ \bar{T}_1)$
$P_{23}\ P_{22}$
$\bar{T}_5\ \bar{T}_3$
$P_{24}-P_{21}$
$P_{19}-P_{16}$
$T_5\ \bar{T}_4\ (\bar{T}_3+\bar{T}_2)$
$P_{14}-P_9$ Sβ
$T_5\ T_4$
P_8-P_3 Sα
$\bar{T}_5\ \bar{T}_4\ (\bar{T}_3+\bar{T}_2)$
$P_{24}-P_{20}$ Tα
$\bar{T}_5\ (\bar{T}_3+\bar{T}_2)$
$P_{24}-P_{15}$
$\bar{T}_3+\bar{T}_2+\bar{T}_1$
$\bar{P}_0$
$T_5+T_4+T_3+T_2+T_1$
$\bar{P}_{24}$
$\bar{T}_5+\bar{T}_4+\bar{T}_2+T_1$
$\bar{P}_5$
EMITTER FOLLOWERS

Figure 4-3. Basic Timing Diagram



$$t_1=\overline{T}_3+\overline{T}_5\ T_4\ S_{ot}+T_5\ \overline{T}_4\ \overline{T}_2+\overline{T}_5\ T_2$$

$$\overline{t}_1=1$$

$$t_2=\overline{T}_3\ T_1+T_5\ T_4\ T_3\ \overline{T}_1$$

$$\overline{t}_2=T_1$$

$$t_3=T_2\ T_1$$

$$\overline{t}_3=T_1\ +\overline{T}_5\ \overline{T}_4\ \overline{T}_2$$

$$t_4\ =T_3\ \overline{T}_1\,(\overline{T}_2\ +T_5\ S_{ot})$$

$$\overline{t}_4\ =T_3\ \overline{T}_2\ \overline{T}_1\ +P_0$$

$$t_5\ =S_{ot}\ (T_3\ T_4\ \overline{T}_2+T_2\ \overline{T}_1\ T_3)$$

$$\overline{t}_5=T_3\ T_2$$

Figure 4-4. Pulse Time Indicator, Logic Diagram

	T_5	T_4	T_3	T_2	T_1
P_{15}	0	1	1	0	0

In order to define P_{14}, the indicator flip-flops must assume the following new states.

	T_5	T_4	T_3	T_2	T_1
P_{14}	1	0	1	0	1

The T_3 and T_2 flip-flops do not require a change of state. However, the state of T_5, T_4, and T_1 must change.

To change the state of T_4, the following logic is required: $\overline{t}_4 = T_3 \overline{T}_2 \overline{T}_1 + P_0$. This equation is satisfied at P_{15} time by the T_1, T_2 and T_3 flip-flops. T_1 and T_2 are low, and T_3 is high; i. e., $T_3 \overline{T}_2 \overline{T}_1$. Therefore, T_4 changes state.

To turn on T_1, the following logic is required:

$$t_1 = \overline{T}_3 + \overline{T}_5 T_4 S_{ot} + T_5 \overline{T}_4 \overline{T}_2 + \overline{T}_5 T_2$$

The state of the pulse time indicator flip-flops at P_{15} allow only the $\overline{T}_5 T_4 S_{ot}$ portion of the OR gate to satisfy the equation. The T_1 flip-flop is now dependent on the condition of the sector address track (S_{ot}).

To turn on T_5, the following logic is required: $S_{ot} T_3 T_4 \overline{T}_2$. The condition of the pulse time indicator flip-flops indicator time P_{15} is such that the $T_3 T_4 \overline{T}_2$ portion of the OR circuit is satisfied. However, the T_5 flip-flop is dependent on the condition of S_{ot}.

The condition of S_{ot} will now dictate the pulse time defined by the indicator. Where a 1 appears in all 64 words of S_{ot} at indicator time P_{15}, the T_1 and T_5 flip-flops will turn on and the indicator condition will be as follows:

T_5	T_4	T_3	T_2	T_1
1	0	1	0	1

The pulse time indicator is then defining P_{14}, and the normal count will continue.

However, where a 0 appears in any word of the S_{ot} track at indicator time P_{15}, the T_1 and T_5 flip-flops will remain off and the condition of the pulse time indicator will be as follows:

T_5	T_4	T_3	T_2	T_1
0	0	1	0	0

This condition defines the P_{20} pulse time. The indicator will now count down from P_{20} through P_{15}, where the condition of S_{ot} will again decide whether the pulse time indicator will continue the count in a normal manner or reset to P_{20} and attempt to sync once more.

The possibility of a "false sync" exists due to the fact that the data on the S_{ot} track, except for P_{15} time, is random with respect to the sync operation. This "false sync" condition is compensated for by the fact that every time P_{15} is defined by the indicator, a 1 on the sector address track is required to continue the count. The true P_{15} time is the only time that S_{ot} will contain a 1 in all 64 words of the track every time it is defined.

4-28. Sigmator Program Delay. Another function of the pulse time indicator is to delay part of the sigmator program from S_{ot} into the sigmator control flip-flops. This is defined by the following logic:

$$t_5 = S_{ot} \, T_2 \, \overline{T}_1 \, T_3$$

$$t_4 = S_{ot} \, T_5 \, \overline{T}_1 \, T_3.$$

This function is described in paragraph 4-69.

4-29. Order Register

The order register is comprised of three flip-flops, 0_1, 0_2, and 0_3, which allow for eight basic orders. Thirty-two additional operations are provided by time sharing the output of the track selection register flip-flops,

S_1 through S_5, with a store order (Sr) to define unique kinds of store orders. This provides a total of 39 operations, of which, 38 are used to provide 13 different commands. Store order codes are listed in section 5 of this manual. Listed below are the eight basic commands.

	0_1	0_2	0_3	ORDERS
At	0	0	0	Clear and Add
Ex	0	0	1	Extract
Mu	0	1	0	Multiply
Tc	0	1	1	Conditional Transfer
Ad	1	0	0	Add
Dv	1	0	1	Divide
Su	1	1	0	Subtract
Sr	1	1	1	Store (Normal)

The order is read from the instruction in the main memory during the IR phase and is stored in the order flip-flops until the next order is read in except for certain modified store orders. The order is read in to 0_3 and shifted through to 0_2 and 0_1, so that the bit at P_2 ends up in 0_1, the bit at P_1 ends up in 0_2, and the bit at P_0 remains in 0_3. Figure 4-5 shows the wave forms of the order register flip-flops as the order is shifted through to its final configuration. The order from the main memory (M_{oy}) is read in to the order register during the IR phase ($K_1 K_2 \overline{K}_3$). The arrows indicate where each bit is stored. The order illustrated by the figure is the divide order.

Figure 4-5 illustrates the logic network for the order register. Logic for the order register flip-flops is listed below.

$$o_1 = (K_1 K_2 \overline{K}_3 + L)\ 0_2$$

$$\overline{o}_1 = (K_1 K_2 \overline{K}_3 + L)\ \overline{0}_2 + 0_2\ 0_3\ \overline{K}_1\ \overline{K}_2\ S_5\ P_0$$

$$o_2 = (K_1 K_2 \overline{K}_3 + L)\ 0_3$$

$$\overline{o}_2 = (K_1 K_2 \overline{K}_3 + L)\ \overline{0}_3 + R_3$$

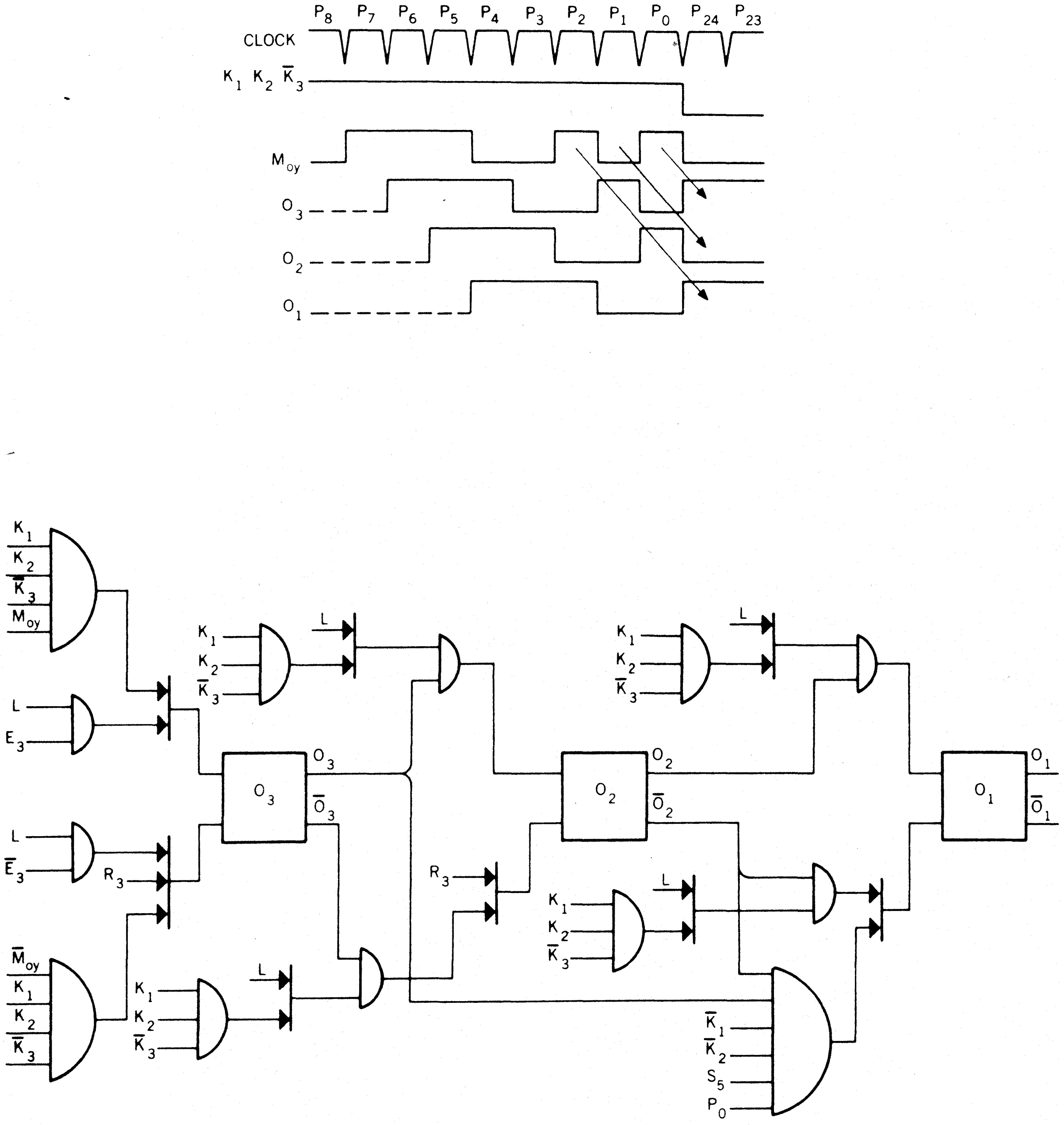


Figure 4-5. Order Register, Logic and Timing Diagram

$$o_3 = K_1 K_2 \overline{K}_3 M_{oy} + L E_3$$

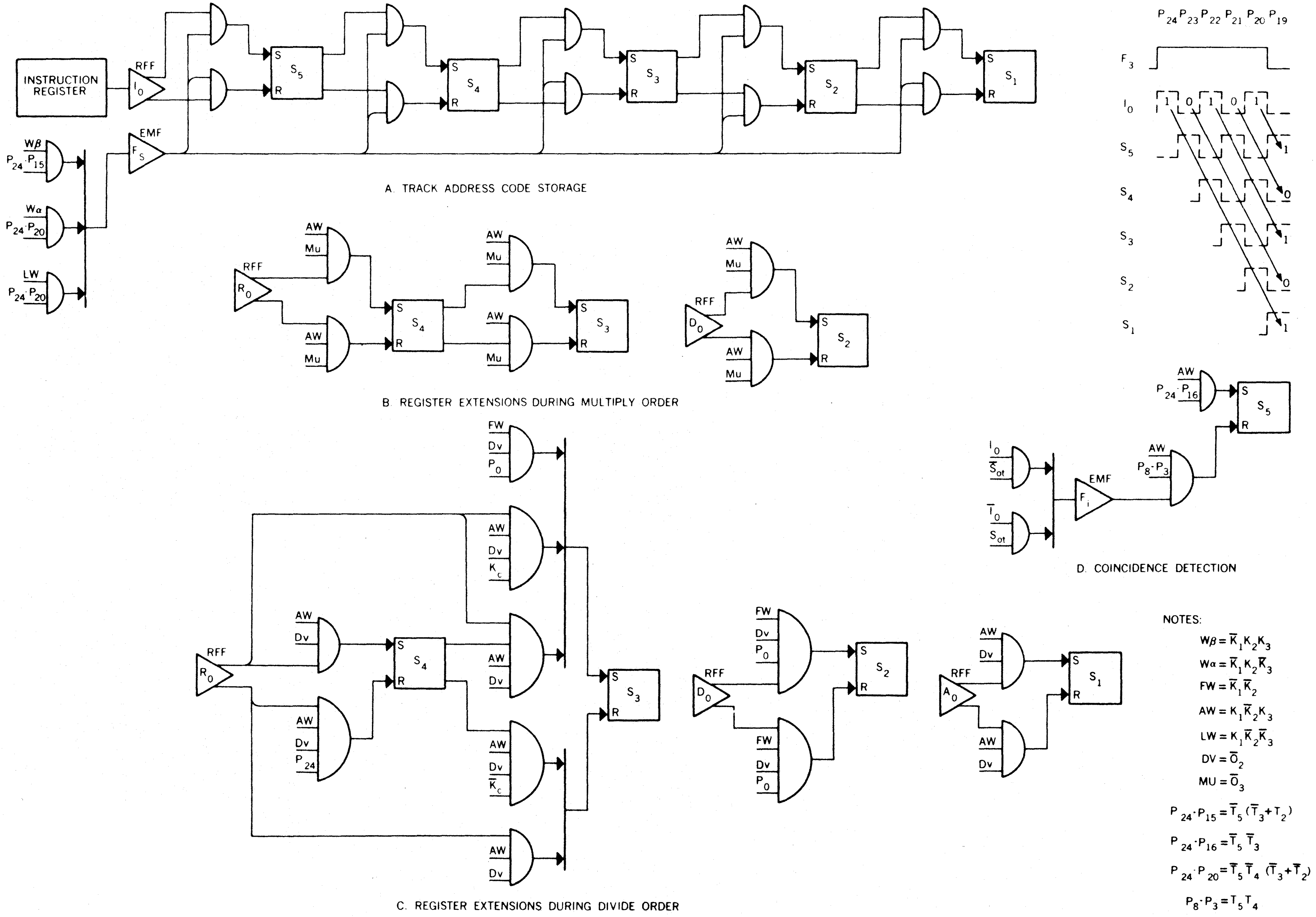
$$\overline{o_3} = K_1 K_2 \overline{K}_3 \overline{M}_{oy} + L \overline{E}_3 + R_3$$

The logic for the order register flip-flops is contained in the computer with the exception of L, E_3, and R_3. The Fill Test Unit supplies L and E_3, with L being the "fill test instruction fill control" which is normally low, and E_3 being the output of the "time-shared data hold and error detecting flip-flop". The external control provides R_3 which is the "computer stop sequence control." This signal is also normally low. When R_3 is high, it sets the order register to $\overline{0}_2 \overline{0}_3$ (Add or Clear and Add). The equation $0_2 0_3 \overline{K}_1 \overline{K}_2 S_5 P_0$ resets 0_1 at P_0 of FW during transition to AW and when S_5 is high; this changes the order from store to Conditional Transfer.

4-30. Track Selection Register

The primary purpose of the track selection register is to store main memory track address codes for selecting tracks in main memory. The stored codes are sampled in the head selecting matrix, and the desired track information is routed from the selected head via the time-shared M_{oy} read flip-flop. During store orders the track selection register stores codes which designate specific store orders. Particular flip-flops of the track selection register are time shared in multiply and divide orders (see parts b and c of figure 4-6). During AW, flip-flop S_5 is used for coincidence detection (see part d of figure 4-6). The track selection register comprises five flip-flops (S_1, S_2, S_3, S_4, and S_5) and associated logic.

Track and store order codes are shifted into and stored in the track selection register from the instruction register I_0 during Wα , Wβ, and LW phases as shown in part a of figure 4-6. The outputs of the track selection register are wired to the head selecting matrix, but are only effectual during IR, FW, and LW phases; that is, during read-in times.



NOTES:

Wβ = $\bar{K}_1K_2K_3$

Wα = $\bar{K}_1K_2\bar{K}_3$

FW = $\bar{K}_1\bar{K}_2$

AW = $K_1\bar{K}_2K_3$

LW = $K_1\bar{K}_2\bar{K}_3$

DV = $\bar{O}_2$

MU = $\bar{O}_3$

$P_{24}\cdot P_{15} = \bar{T}_5(\bar{T}_3+T_2)$

$P_{24}\cdot P_{16} = \bar{T}_5\bar{T}_3$

$P_{24}\cdot P_{20} = \bar{T}_5\bar{T}_4(\bar{T}_3+\bar{T}_2)$

$P_8\cdot P_3 = T_5T_4$

Figure 4-6. Track Selection Register, Logic and Timing Diagram

The coding of the main memory track addresses employs the natural binary number system with flip-flop S_1 representing the least significant bit. For example, the code for head 24 is $\overline{S}_1\ \overline{S}_2\ \overline{S}_3\ S_4\ S_5$. Heads 4 through 15 and 20 through 31 are fully coded by the track selection register independent of relay S_6. Heads 0, 1, 2, 3, 16, 17, 18, and 19 are coded by the natural binary number system of the track selection register plus the condition that $\overline{S}_6$ is high. Heads $\underline{0}$, $\underline{1}$, $\underline{2}$, $\underline{3}$, $\underline{16}$, $\underline{17}$, $\underline{18}$, and $\underline{19}$ are also coded by the natural binary number system of the track selection register but with S_6 high.

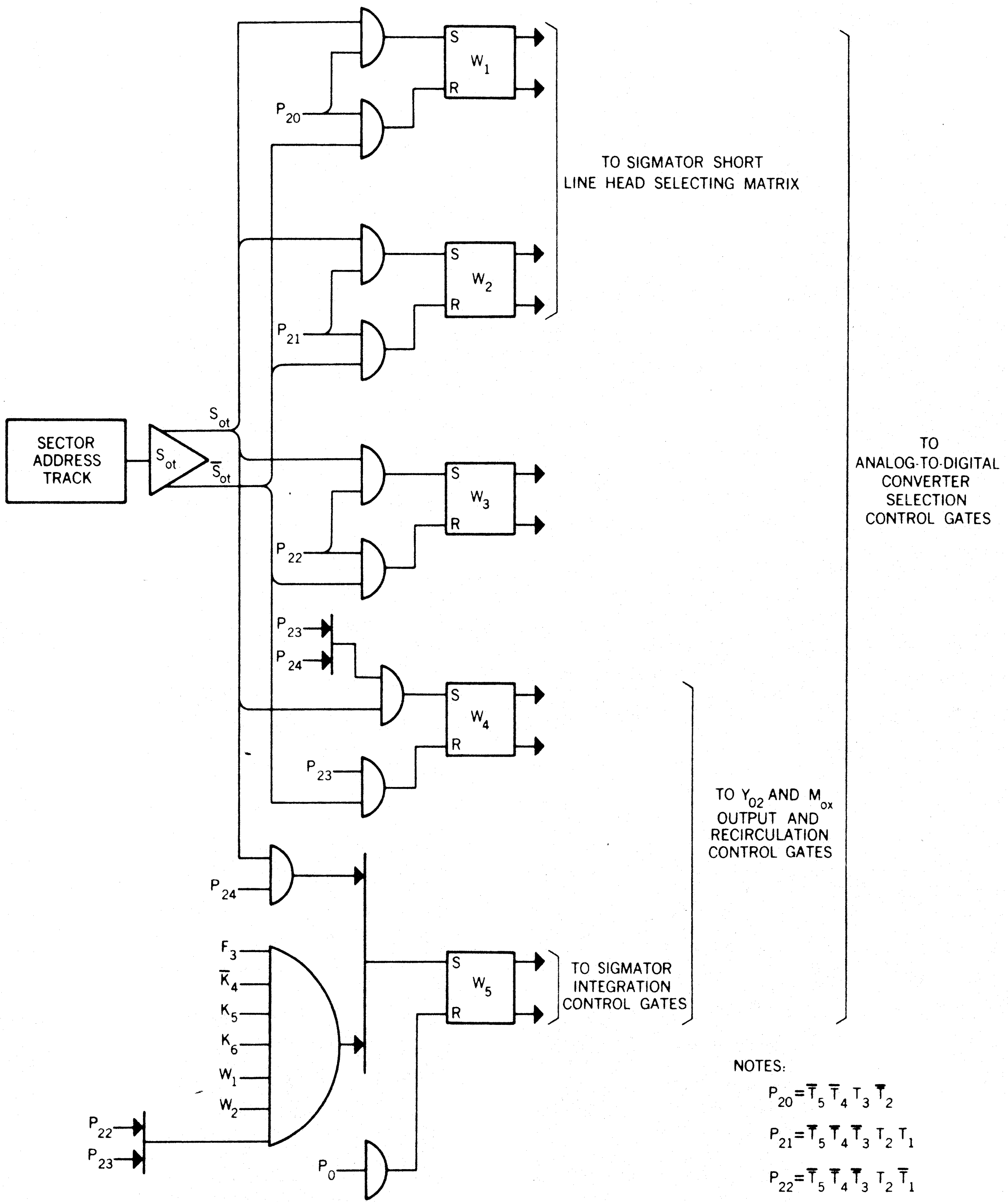
Track selection register information is interpreted as store order qualifier codes when the order register contains the store order code ($0_1\ 0_2\ 0_3$) during the FW phase or the Tc order code ($\overline{0}_1\ 0_2\ 0_3$) during the AW phase. These codes which are without binary significance, are listed in section 5 of this manual.

For detailed explanation on the use of flip-flops S_1, S_2, S_3, and S_4 during multiplication and division orders, refer to paragraphs 4-67 and 4-68, respectively. For details on the use of flip-flop S_5 for coincidence detection during AW, refer to paragraphs 4-66, 4-67, and 4-68.

4-31. Input-Output Selection Register

The input-output selection register is used to store program codes for one word time periods. These codes are used (1) to select analog-to-digital converters in external control for input and output functions, (2) to select output information from M_{ox} or Y_{o2}, or Ø1 time word, (3) to select sigmator short line readheads Δ1, Δ2, Δ3, or ΔT via the sigmator head selecting matrix and read flip-flop X_{o2}, and (4) to control recirculation of M_{ox}.

The input-output selection register comprises five flip-flops (W_1, W_2, W_3, W_4, and W_5) and their associated logic. As shown in figure 4-7, the input output selection register is filled during P_{24} through P_{20} of


S
W1
R
P20
TO SIGMATOR SHORT LINE HEAD SELECTING MATRIX
S
W2
R
P21
SECTOR ADDRESS TRACK
Sot
Sot
S̄ot
S
W3
R
P22
TO ANALOG-TO-DIGITAL CONVERTER SELECTION CONTROL GATES
P23
P24
S
W4
R
P23
TO Y02 AND Mox OUTPUT AND RECIRCULATION CONTROL GATES
P24
F3
K̄4
K5
K6
W1
W2
S
W5
R
TO SIGMATOR INTEGRATION CONTROL GATES
P22
P23
P0

NOTES:

$P_{20} = \overline{T}_5 \overline{T}_4 T_3 \overline{T}_2$

$P_{21} = \overline{T}_5 \overline{T}_4 \overline{T}_3 T_2 T_1$

$P_{22} = \overline{T}_5 \overline{T}_4 \overline{T}_3 T_2 \overline{T}_1$

$P_{23} = \overline{T}_5 \overline{T}_4 \overline{T}_3 \overline{T}_2 T_1$

$P_{24} = \overline{T}_5 \overline{T}_4 \overline{T}_3 \overline{T}_2 \overline{T}_1$

Figure 4-7. Input-Output Selection Register, Logic and Timing Diagram

each word time by the programmed codes on the sector selection register S_{ot}. Flip-flop W_5 is time shared with the sigmator [logic term $F_3\overline{K}_4K_5K_6W_1W_2$ $(P_{23} + P_{22})$] for the purpose of allowing or deleting integrations.

The input-output selection register codes are provided in table 4-1.

Table 4-1

Input-Output Selection Register Codes

W_5	W_4	W_3	W_2	W_1	Sigmator Head	Output	Converter Address	Other
0	0	0	0	0	Δ 1	M_{ox}	0	M_{ox} (recirculates)
0	0	0	0	1	Δ 2	M_{ox}	1	M_{ox} (recirculates)
0	0	0	1	0	Δ 3	M_{ox}	2	M_{ox} (recirculates)
0	0	0	1	1	Δ T	M_{ox}	3	M_{ox} (recirculates)
0	0	1	0	0	Δ 1	M_{ox}	4	M_{ox} (recirculates)
0	0	1	0	1	Δ 2	M_{ox}	5	M_{ox} (recirculates)
0	0	1	1	0	Δ 3	M_{ox}	6	M_{ox} (recirculates)
0	0	1	1	1	Δ T	M_{ox}	7	M_{ox} (recirculates)
0	1	0	0	0	Δ 1	Y_{o2}	8	M_{ox} (recirculates)
0	1	0	0	1	Δ 2	Y_{o2}	9	M_{ox} (recirculates)
0	1	0	1	0	Δ 3	Y_{o2}	10	M_{ox} (recirculates)
0	1	0	1	1	Δ T	Y_{o2}	11	M_{ox} (recirculates)
0	1	1	0	0	Δ 1	Y_{o2}	12	M_{ox} (recirculates)
0	1	1	0	1	Δ 2	Y_{o2}	13	M_{ox} (recirculates)
0	1	1	1	0	Δ 3	Y_{o2}	14	M_{ox} (recirculates)
X	1	1	1	1	Δ T	$\emptyset_1$ Time Word		M_{ox} (recirculates)

Table 4-1 (continued)

Input-Output Selection Register Codes

W_5	W_4	W_3	W_2	W_1	Sigmator Head	Output	Converter Address	Other
1	0	0	0	0	Δ 1	M_{ox}	16	M_{ox}(non recirculate)
1	0	0	0	1	Δ 2	M_{ox}	17	M_{ox}(non recirculate)
1	0	0	1	0	Δ 3	M_{ox}	18	M_{ox}(non recirculate)
1	0	0	1	1	Δ T	M_{ox}	19	M_{ox}(non recirculate)
1	0	1	0	0	Δ 1	M_{ox}	20	M_{ox}(non recirculate)
1	0	1	0	1	Δ 2	M_{ox}	21	M_{ox}(non recirculate)
1	0	1	1	0	Δ 3	M_{ox}	22	M_{ox}(non recirculate)
1	0	1	1	1	Δ T	M_{ox}	23	M_{ox}(non recirculate)
1	1	0	0	0	Δ 1	Y_{o2}	24	M_{ox}(recirculates)
1	1	0	0	1	Δ 2	Y_{o2}	25	M_{ox}(recirculates)
1	1	0	1	0	Δ 3	Y_{o2}	26	M_{ox}(recirculates)
1	1	0	1	1	Δ T	Y_{o2}	27	M_{ox}(recirculates)
1	1	1	0	0	Δ 1	Y_{o2}	28	M_{ox}(recirculates)
1	1	1	0	1	Δ 2	Y_{o2}	29	M_{ox}(recirculates)
1	1	1	1	0	Δ 3	Y_{o2}	30	M_{ox}(recirculates)
X	1	1	1	1	Δ T	$\emptyset_1$ Time Word		M_{ox}(recirculates)

4-32. Phase Sequencing

The phase sequencing control provides overall sequential control of operations. The various phases normally follow one another in a limited number of fixed sequential patterns as shown in figure 4-8. Each phase is individually discussed in paragraphs 4-33 through 4-39.

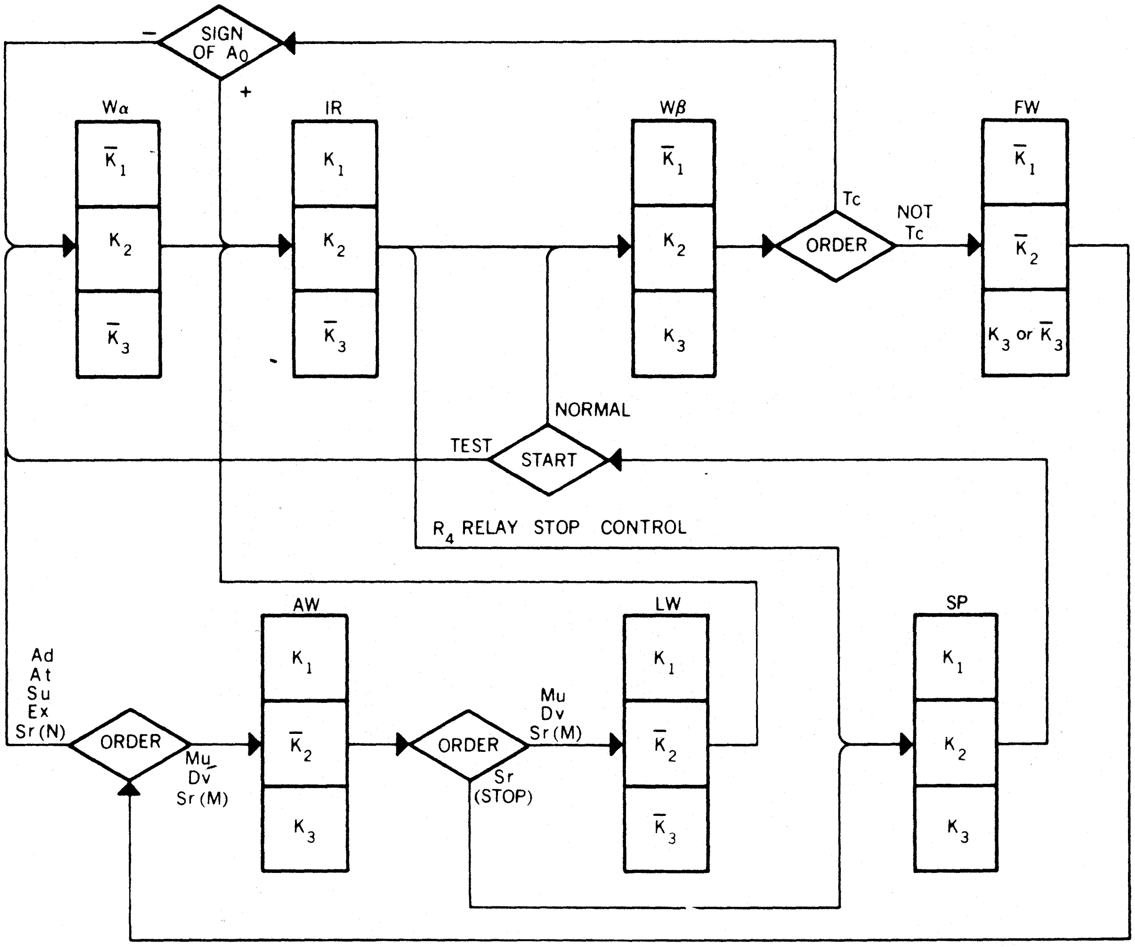


Figure 4-8. Phase Sequencing Diagram

4-33. Wait Alpha Phase (Wα) $\overline{K}_1K_2\overline{K}_3$. During wait alpha phase, a search is made for the next instruction to be performed. The search consists of a comparison of the sector address portion (Sα) of the present instruction in the instruction register I_0 with the sector address track S_{ot}. When I_0 equals S_{ot}, control signals are generated which pass the computer into phase 2.

The computer enters wait alpha phase when the K flip-flops are in the following states: $K_1 = 0$, $K_2 = 1$, and $K_3 = 0$ or, in logical terms, W $= \overline{K}_1K_2\overline{K}_3$. Wait alpha is entered from either phase wait beta, first word, or stop. When phase wait alpha ends, after an indefinite number of word times, the computer unconditionally enters the instruction read-in phase IR.

Detailed description of wait alpha phase is as follows: The K_c flip-flop is set at P_{23} of each word time by $\overline{K}_1K_2\overline{T}_5\overline{T}_3$ as shown in figure 4-9. During pulse times P_8 through P_3 of each word time I_0/S_{ot} sector address comparison is made. If any of the bits of I_0 and S_{ot} are not identical, a high signal is passed by one of the comparator AND gates to emitter follower F_i. The high F_i output resets K_c thereb preventing K_1 from being set. Thus the computer is forced to remain in phase 1 for another word time. This type of event is shown in the first word time illustrated in figure 4-9.

During the word time that I_0 equals S_{ot} in the P_8 through P_3 interval the comparator AND gates remain low. Thus F_i remains low and K_c cannot reset. At P_0 time, therefore, K_c is high to set K_1 at P_0 thus terminating wait alpha phase. This type event is shown in the second word time illustrated in figure 4-9.

4-34. Instruction Read-In Phase (IR) $K_1K_2\overline{K}_3$. The instruction read-in phase is always one word time long and is the same for all orders. During the IR phase, a selected instruction word from main memory is read into the instruction register I_0 and concurrently shifted into the order register via the M_{oy} read flip-flop. See figure 4-10. The track address of the instruction word is defined by the track address register which received the track address code during the previous phase.

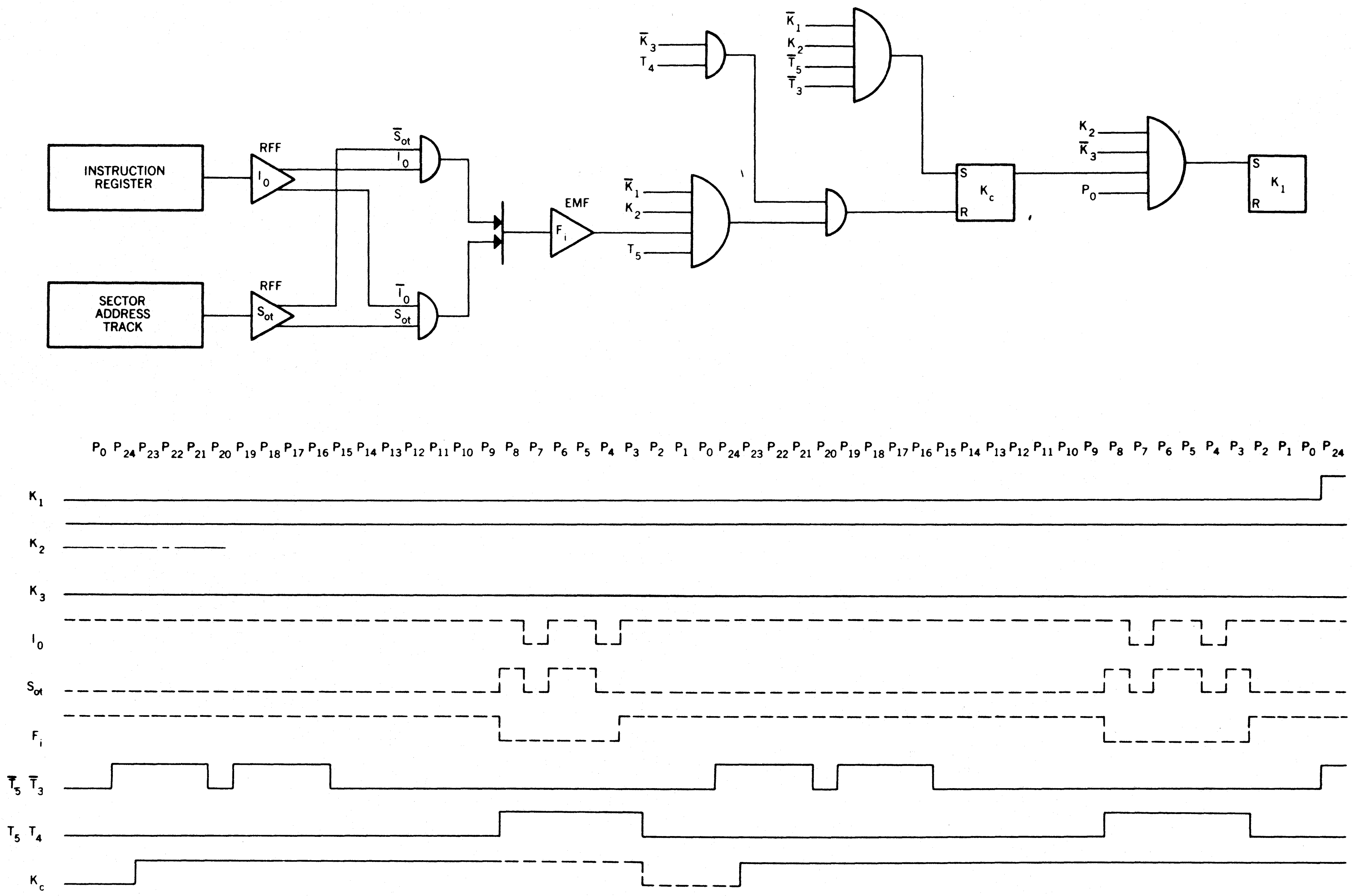
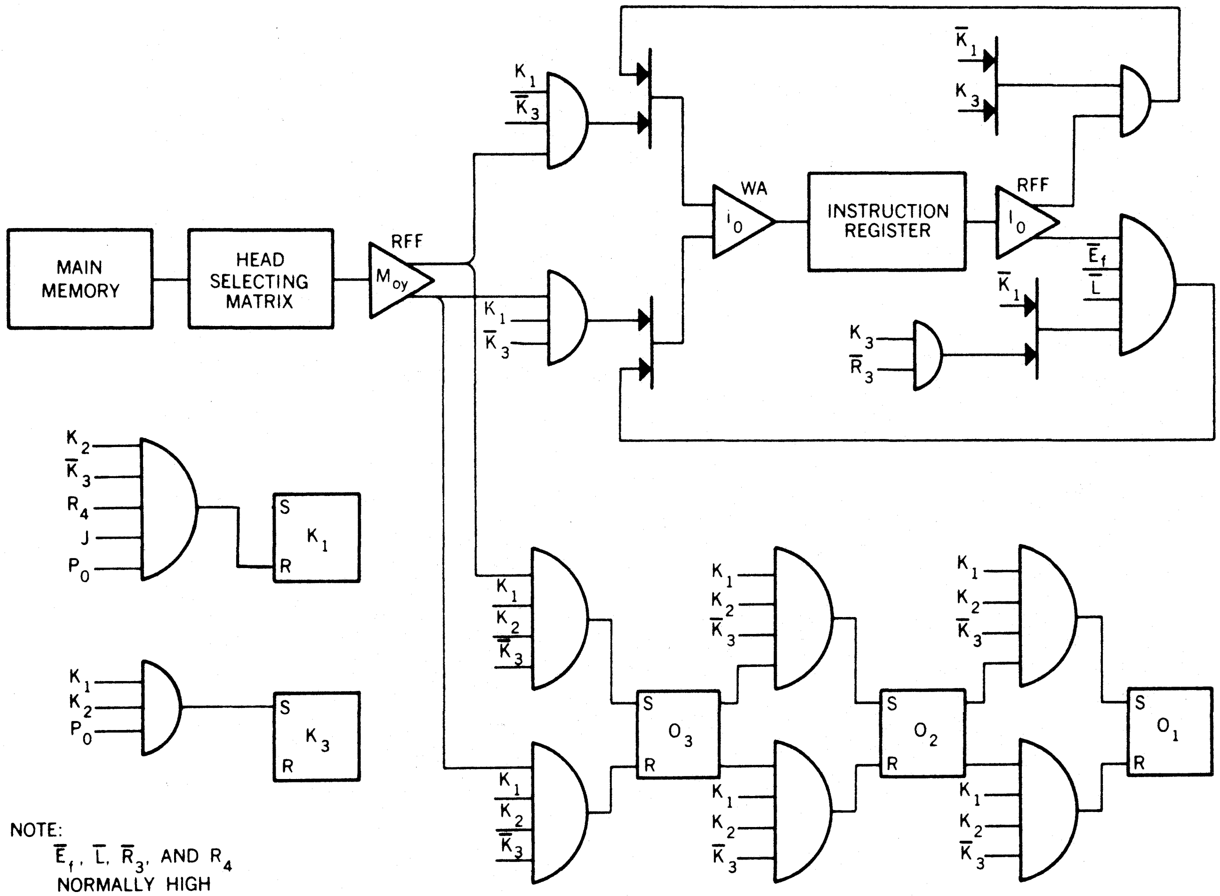

INSTRUCTION REGISTER
SECTOR ADDRESS TRACK
RFF
I_0
RFF
S_{ot}
$\overline{S}_{ot}$
I_0
$\overline{I}_0$
S_{ot}
EMF
F_i
$\overline{K}_3$
T_4
$\overline{K}_1$
K_2
$\overline{T}_5$
$\overline{T}_3$
$\overline{K}_1$
K_2
T_5
S
K_c
R
K_2
$\overline{K}_3$
P_0
S
K_1
R
P_0 P_{24} P_{23} P_{22} P_{21} P_{20} P_{19} P_{18} P_{17} P_{16} P_{15} P_{14} P_{13} P_{12} P_{11} P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0 P_{24} P_{23} P_{22} P_{21} P_{20} P_{19} P_{18} P_{17} P_{16} P_{15} P_{14} P_{13} P_{12} P_{11} P_{10} P_9 P_8 P_7 P_6 P_5 P_4 P_3 P_2 P_1 P_0 P_{24}
K_1
K_2
K_3
I_0
S_{ot}
F_i
$\overline{T}_5$ $\overline{T}_3$
T_5 T_4
K_c

Figure 4-9. Wait Alpha, Logic and Timing Diagram



NOTE:
$\overline{E}_f$, $\overline{L}$, $\overline{R}_3$, AND R_4
NORMALLY HIGH

Figure 4-10. Instruction Read-In, Logic Diagram

The IR phase may be entered from the Wα, the Wβ, or the LW phase. The IR phase is normally entered from Wα following sector coincidence; from Wβ during a conditional transfer order when the sign of the accumulator is positive (refer to paragraph 4-64); and from LW at the completion of multiply, divide, and modified store orders (refer to paragraphs 4-67, 4-68, and 4-66, respectively). When the IR phase is completed, the computer normally proceeds to the Wβ phase. An exception to this occurs when R_4 is held low causing the K_1 flip-flop to remain high. This sends the computer into the stop phase (SP).

The logic for the transition for IR to Wβ is: $\overline{k}_1 = P_0K_2\overline{K}_3JR_4$, and $k_3 = P_0K_1K_2$. The signals J and R_4 are high during the run mode.

4-35. Wait Beta Phase (Wβ) $\overline{K}_1K_2K_3$. During the wait beta phase, a search is made for the operand by comparing the β_s portion of the instruction with the number in the sector address track (S_{ot}). When the numbers are the same (coincidence), the operand is contained in the next word of the track specified by the β track number (βt). After the Wβ phase, the K_2 flip-flop is reset and the first word phase (FW) is entered. The K_2 flip-flop is reset by the following logic: $\overline{k}_2 = J\,P_0\overline{K}_1K_3\,(0_1 + \overline{0}_2 + \overline{0}_3)\,K_c$. At coincidence ($K_cP_0$) goes from W$\beta$ to FW for all orders except conditional transfer (Tc).

When the conditional transfer takes place, the Wβ state goes to wait alpha (Wα) at P_{24} of the first Wβ word, if the contents of the accumulator are negative. That is, when a conditional transfer order exists and the sign of the accumulator contents is negative, the K_3 flip-flop is reset and Wβ goes to Wα at P_{24}. The K_3 flip-flop is reset by the following logic: $\overline{k}_3 = A_1P_{24}\overline{K}_1\overline{0}_10_20_3$. When A_0 is positive, the β address specifies the address of the next instruction instead of an operand. That is, when a Tc order exists and A_0 is positive, K_1 is set and K_3 is reset. At coincidence (K_cP_0) Wβ goes to IR. The K_1 flip-flop is set by the following logic: $k_1 = P_0K_2\,(\overline{0}_10_20_3)\,K_c$, while the K_3 flip-flop is reset as follows: $\overline{k}_3 = P_0J\overline{K}_1K_2K_c$.

Another exception to the rule exists when the operation is a store order (Sr). During this condition the β address either specifies where the contents of the accumulator are to be stored in the main memory, or one of the several modified store commands. In either case the Wβ phase goes on to FW at P_0.

4-36. First Word Phase (FW) $\overline{K}_1\overline{K}_2$ (K_3 or $\overline{K}_3$). The first word phase is one word time in duration. In this phase, the operation specified by the order code is begun. All clear and add (At), add (Ad), subtract (Su), extract (Ex), and normal store operations are completed in this phase. After completion of any of the operations listed above, the K_2 flip-flop is set by the following logic: $k_2 = \overline{K}_1 P_0 J \;\; \overline{0}_1 0_3 + 0_1 0_2 \overline{S}_5 \, (\overline{S}_4 + \overline{S}_3) + \overline{0}_1 \overline{0}_2$. This sends FW to Wα at P_0. When the order specified is multiply (Mu), divide (Dv), or one of the modified store commands, K_1 and K_3 are set thereby sending the FW phase to additional words (AW). The K_1 flip-flop is set by the following logic: $k_1 = P_0 \overline{K}_2 \;\; \overline{0}_1 0_2 + 0_1 0_3 (S_5 + S_4 S_3) + 0_1 \overline{0}_2 0_3$, while K_3 is set as follows: $k_3 = P_0 \overline{K}_1 \overline{K}_2 \; 0_1 0_3 (S_5 + S_4 S_3 + \overline{0}_2) \;\; + \overline{K}_1 \overline{K}_2 \overline{0}_1 0_2 \; (A_o + P_0)$.

4-37. Additional Words Phase (AW) $K_1\overline{K}_2K_3$. In the additional words phase, multiplication and special operations of the modified store order are completed. All but the last step of the division problem is completed in this phase also. When coincidence occurs (S_5P_0) K_3 is reset and the computer goes from AW to LW. The K_3 flip-flop is reset by the following logic: $\overline{k}_3 = P_0 J K_1 \overline{K}_2 S_5$.

The last word (LW) phase is bypassed when the stop command is given in the form of a modified store order. That is 0_1 is reset by S_5 at P_0 of FW during transition to AW, and the logical equation ($k_2 = P_0 K_1 \overline{0}_1 0_3 \overline{S}_4 \overline{S}_3 \overline{S}_2 \overline{S}_1$) sets the K_2 flip-flop sending the computer into the stop phase (SP).

4-38. Last Word Phase (LW) $K_1\overline{K}_2\overline{K}_3$. In this phase division is completed and the quotient is transferred between registers. The LW phase lasts for one word time. The K_2 flip-flop is set at P_0 of LW by the following logic: $k_2 = P_0 K_1 \overline{K}_3$. This action returns the computer to the IR phase. The LW phase is entered from the AW phase.

4-39. Stop Phase (SP) $K_1K_2K_3$. In this phase only the sigmator continues to compute and if the control switch is in the off position, electrical power is removed from the computer.

The stop phase is entered into from the AW phase (paragraph 4-39) or the IR phase (paragraphs 4-57 and 4-58. After the stop phase two types of starts may be used. The normal start sequence where the computer enters the Wβ phase by resetting the K_1 flip-flop, or the fill test start in which K_1 and K_3 are reset. For the normal start sequence K_1 is reset by the following logic: $\overline{k}_1 = S_{ot}P_{18}R_1$. A control sequence relay provides R_1 and $S_{ot}P_{18}$ is a normal program start. For the fill test start, K_1 is reset by the following logic: $\overline{k}_1 = JP_0K_2R_4F$; K_3 is reset by the following logic: $\overline{k}_3 = P_0JK_1K_2MdF$. Signal F is a fill-test start signal, R_4 is provided by a control sequence relay, and Md is the signal designating Wα time from the fill-test unit.

4-40. Serial Adder

The serial adder and its carry control flip-flop performs three basic operations. These operations are:

Add	$A_1 + D_1 = a_0$
Normal Subtract	$A_1 - D_1 = a_0$
Abnormal Subtract	$D_1 - A_1 = a_0$

Normal subtraction is used in the multiplication and division orders and is accomplished during the AW phase. Abnormal subtraction is used for subtraction during the FW phase. The state of the carry control flip-flop (K_c) determines whether C_a is to operate with carry logic (addition) or borrow logic (subtraction). When the carry control flip-flop is low ($\overline{K}_c$), C_a operates with carry logic, and an add operation takes place. When K_c is high, Ca operates with borrow logic and a subtraction operation takes place. The carry control flip-flop is set to its proper state at P_0 of the Wβ phase prior to FW.

The operation of the serial adder is determined by the applied logic. Figure 4-11 illustrates the logic circuitry for the serial adder. The logic for the add operation is divided into three steps. These steps are:

1. $c_a = \overline{K}_c A_1 D_1 \overline{P}_{24}$

   $\overline{c}_a = \overline{K}_c \overline{A}_1 \overline{D}_1 + P_{24}$

2. $f_a = C_a \overline{A}1 + \overline{C}_a A_1 F_b$

   $\overline{f}_a = C_a A_1 + \overline{C}_a \overline{A}_1 F_b$

3. $a_0 = \overline{E}_f \; (F_a \overline{D}_1 + \overline{F}_a D_1)$

   $\overline{a}_0 = \overline{E}_f \; (F_a D_1 + \overline{F}_a \overline{D}_1)$

In step 1 the carry flip-flop (C_a) output is determined. The carry control flip-flop (K_c) is low for the add operation by the following logic: $\overline{k}_c = \overline{K}_1 K_2 \overline{0}_2 P_0$. The states of the A_1 and D_1 flip-flops are determined by the A_0 read flip-flop and the M_{oy} read flip-flop respectively. The A_1 flip-flop copies A_0 during FW while the D_1 flip-flop copies M_{oy}.

In step 2 the output of the A_1 flip-flop and the carry flip-flop are used to control the F_a and $\overline{F}_a$ emitter followers. When the outputs of A_1 and C_a are not of the same state, the F_a output is high. When the outputs of A_1 and C_a are both high or both low, the output of the $\overline{F}_a$ emitter follower is high. Whenever the $\overline{C}_a$ output is employed with the A_1 or $\overline{A}_1$ output to control the F_a or $\overline{F}_a$ emitter followers, the F_b output of the normal serial adder qualifier emitter follower is added to the input logic.

In step 3 the signal $\overline{E}_f$ is normally high. The outputs F_a or $\overline{F}_a$ and D_1 or $\overline{D}_1$ provide input logic for the A_0 write amplifier. When D_1 and F_a are unlike, a_0 is high. When D_1 and F_a are alike, a_0 is produced.

The contents of the accumulator (A_0) and the number contained in the main memory (M_{oy}) are added together during FW and stored in the A_0 register. Addition is performed as follows:

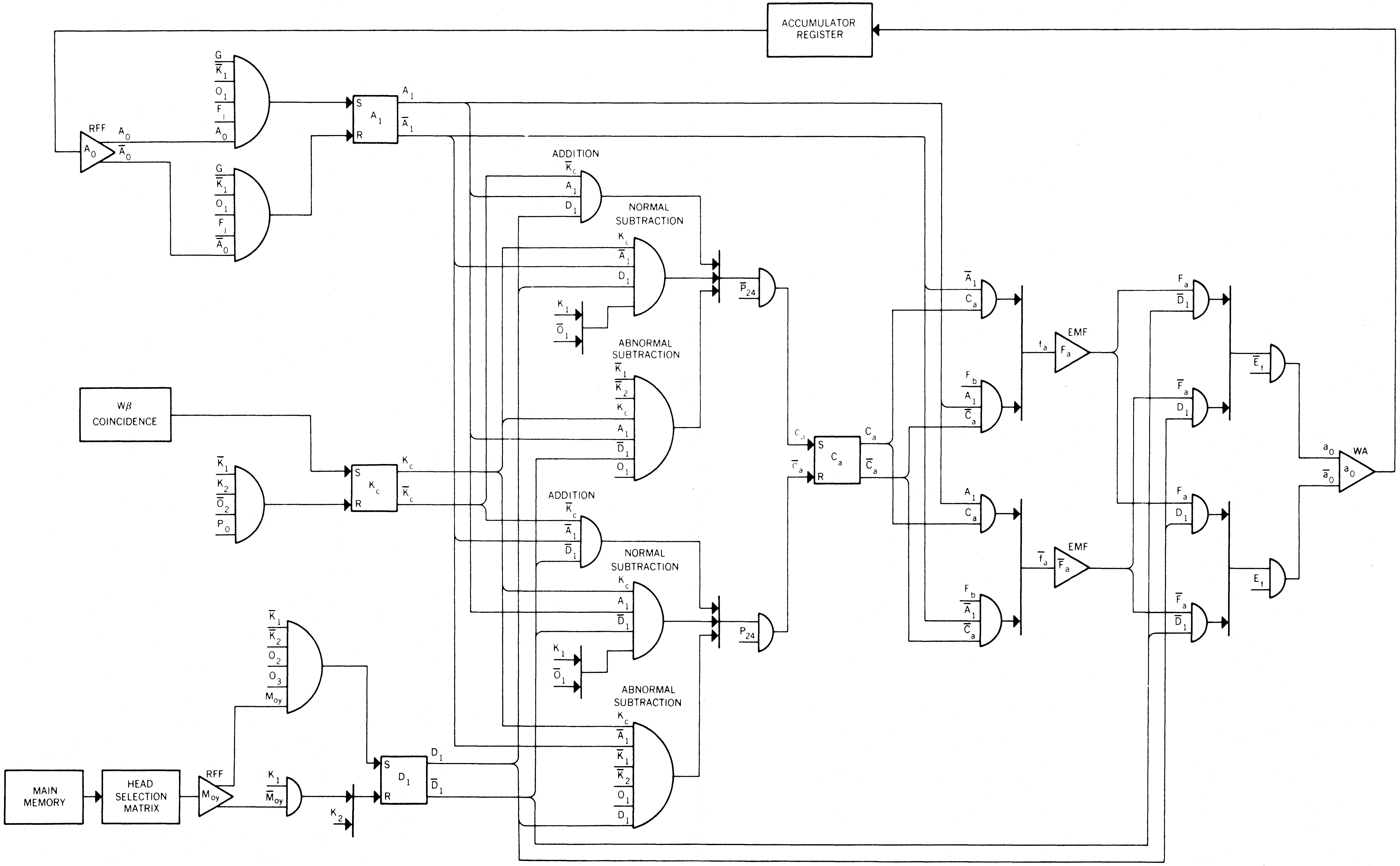


Figure 4-11. Serial Adder, Logic Diagram

$$\begin{array}{ll}
 & P_{23} \\
 & \text{LSD} \\
C_a & 0\ 1\ 0\ 0\ 1\ 1\ 1\ 1\ 1 \\
A_1 & 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \\
D_1 & 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1 \\
f_a & 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\
\hline
a_0 & 0\ 1\ 1\ 0\ 0\ 0\ 1\ 0
\end{array}$$

The normal subtraction ($A_1 - D_1 = a_0$) which is used in the multiplication and division orders require much the same logic as the addition operation. Steps 2 and 3 of the add logic are used in both of the subtraction operations of the serial adder. However, the first step is changed somewhat in that $c_a = K_c \overline{A}_1 D_1 \overline{P}_{24} (K_1 + \overline{0}_1)$, and $\overline{c}_a = K_c A_1 \overline{D}_1 (K_1 + \overline{0}_1) + P_{24}$. Normal subtraction is performed in the following manner:

$$\begin{array}{ll}
 & P_{23} \\
 & \text{LSD} \\
C_a & 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0 \\
A_1 & 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1 \\
D_1 & 0\ 1\ 1\ 0\ 1\ 0\ 0\ 1 \\
f_a & 1\ 1\ 0\ 0\ 1\ 1\ 0\ 1 \\
\hline
a_0 & 1\ 0\ 1\ 0\ 0\ 1\ 0\ 0
\end{array}$$

The abnormal subtraction ($D_1 - A_1 = a_0$) employs the same logic for steps 2 and 3 as does the add and normal subtract operations of the serial adder. In the first step of this operation the logic is: $c_a = \overline{K}_1 \overline{K}_2 K_c A_1 \overline{D}_1 0_1 \overline{P}_{24}$, and $\overline{c}_a = \overline{K}_1 \overline{K}_2 K_c 0_1 \overline{A}_1 D_1 + P_{24}$. Abnormal subtraction is performed in the following manner:

$$\begin{array}{ll}
 & P_{23} \\
 & \text{LSD} \\
C_a & 0\ 0\ 0\ 1\ 1\ 1\ 0\ 0\ 0 \\
D_1 & 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1 \\
A_1 & 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0 \\
f_a & 1\ 0\ 1\ 0\ 0\ 1\ 1\ 0 \\
\hline
a_0 & 0\ 0\ 1\ 1\ 0\ 0\ 0\ 1
\end{array}$$

In both normal and abnormal subtraction when the minuend is zero and the subtrahend is one, the carry is set high. Where the minuend is one and the subtrahend is zero, the carry is reset. Where both numbers are alike, the carry remains in its previous state.

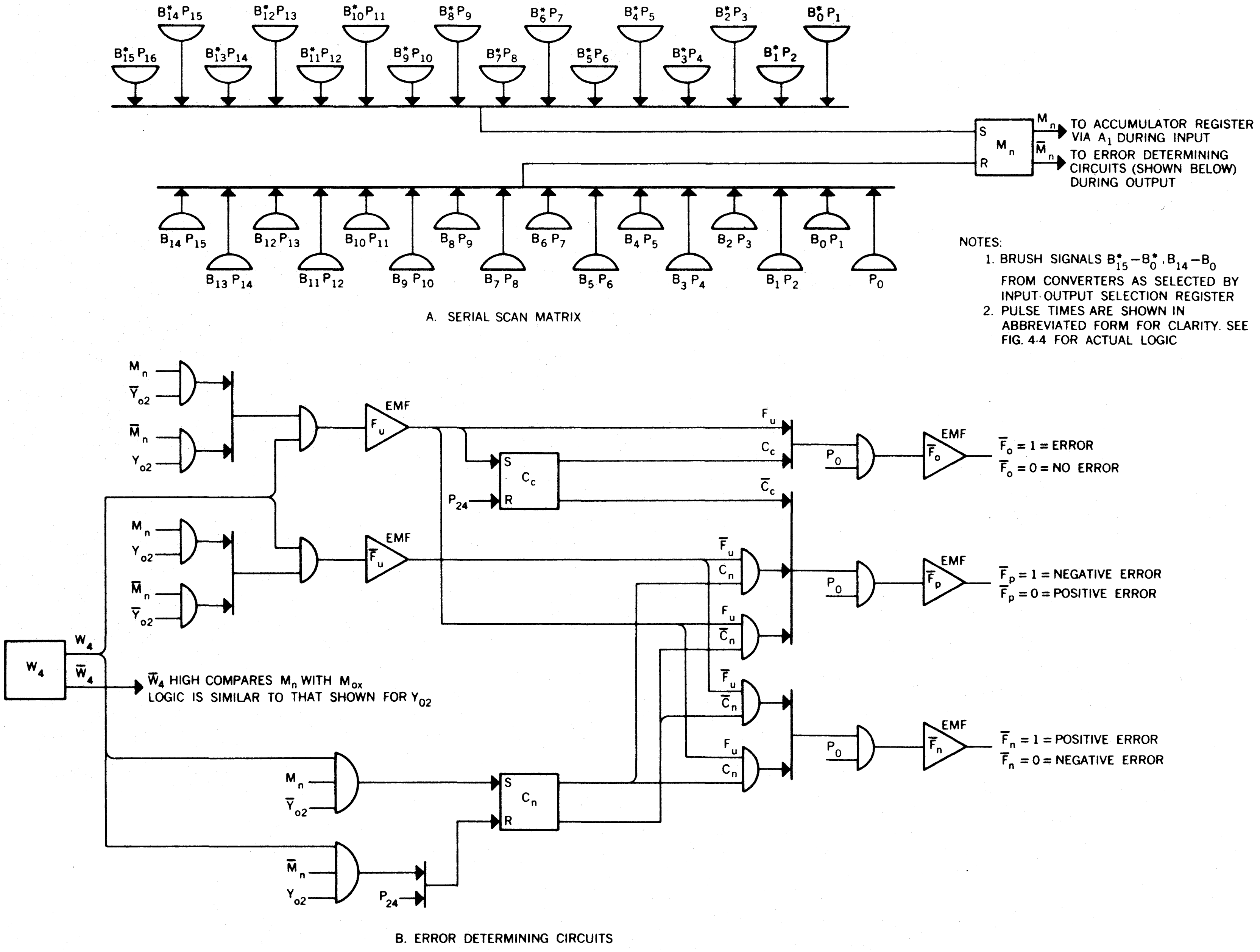
4-41. INPUT-OUTPUT OPERATIONS

Converter scan flip-flop M_n receives brush signals from externally located analog-to-digital converters as selected by input-output selection register codes. Refer to paragraph 4-31. When a particular converter is selected, a +35V DC signal is passed to the converter to sample its binary-coded discs through the agency of associated brushes. Brushes which are in contact with conductive segments of the code discs pass the 35 volt signal to the scan matrix to produce a parallel binary representation of the shaft value of the converter. The serial scan matrix samples this parallel binary representation which is then copied by flip-flop M_n.

During input operations the output of M_n is transferred to the accumulator register via flip-flop A_1. During output operations, the output of M_n is sent to the error determining circuits which compare M_n with either the general purpose output M_{ox} or the sigmator output Y_{o2}. Difference or error signals are generated by the error determining circuits and are used to drive the converter shaft in a direction to reduce the error; that is, to make the converter value approach the value of M_{ox} or Y_{o2}. The error signals control converter shaft positions by means of a servo amplifier, servomotor, and gear train associated with the converter.

4-42. Scan Matrix

In the scanning process, flip-flop M_n is initially reset at P_0 time, as shown in part a of figure 4-12. It remains reset until the occurrence of 1, the first high starred brush signal (the first one possible occurs at P_{16} time if B^*_{15}, the least significant bit, is high). When M_n is



A. SERIAL SCAN MATRIX

NOTES:

1. BRUSH SIGNALS $B^*_{15}-B^*_0$, $B_{14}-B_0$ FROM CONVERTERS AS SELECTED BY INPUT-OUTPUT SELECTION REGISTER
2. PULSE TIMES ARE SHOWN IN ABBREVIATED FORM FOR CLARITY. SEE FIG. 4-4 FOR ACTUAL LOGIC

B. ERROR DETERMINING CIRCUITS

Figure 4-12. Input and Output Operations, Logic Diagram

set, the next high unstarred brush signal (0) will reset M_n. The scanning process continues in this manner until the sign digit is sampled at P_1time. The one-bit delay of the M_n flip-flop brings the converter number into correspondence timewise with other computer data; that is, the sign bit occurs at P_0 time.

4-43. Error Determining Circuits

During output operations the shaft value of a converter as represented by M_n is compared with the desired output value of either M_{ox} or Y_{o2}. When $M_n \neq M_{ox}$ or Y_{o2}, error determining output signals $\overline{F}_o$, $\overline{F}_p$, and $\overline{F}_n$ indicate the error and its direction (either positive or negative) which are used to drive the converter shaft in the direction which are used to drive the converter shaft in the direction which reduces the error. When $M_n = M_{ox}$ or Y_{o2}, the equality, or no-error condition, is indicated by $\overline{F}_o$, $\overline{F}_p$, and $\overline{F}_n$ and the position of the converter shaft is not altered.

Flip-flops C_c and C_n (see part b of figure 4-12) are initially reset by the P_{24} pulse and remain reset until the comparison of M_n with M_{ox} or Y_{o2} begins at P_{15} or later. Throughout the remaining discussion Y_{o2} will be used as an example; however whatever is said of Y_{o2} applies equally to M_{ox}. Flip-flop C_c is used to store a difference in numerical value of M_n and Y_{o2}. During any bit time should a difference exist, emitter follower F_u will go high and set C_c. Flip-flop C_c would then remain set for the remainder of the word time. If there is no numerical difference between M_n and Y_{o2}, F_u will not go high and C_c will be in its reset condition at P_0 time when it is sampled by the error emitter follower $\overline{F}_o$. The F_u signal bypasses C_c to skirt the one-bit delay of the flip-flop, so that the signs of M_n and Y_{o2} may also be sampled at P_0 time. If any difference (either numerical or sign) exists between M_n and Y_{o2}, $\overline{F}_o$ will go high at P_0 time to indicate an error. If no difference exists, $\overline{F}_o$ will be low at P_0 time to indicate no-error.

While M_n and Y_{o2} are being compared to detect a difference by F_u, C_c, and $\overline{F}_o$ logic, the C_n logic is comparing the magnitudes of the two numerical values. Flip-flop C_n is set whenever $M_n > Y_{o2}$, and reset whenever $M_n < Y_{o2}$. Therefore, at P_0 time the output of C_n will be set if the numerical value of $M_n > Y_{o2}$, or reset if the numerical value of $M_n < Y_{o2}$. The outputs of C_n are compared at P_0 time with F_u (unlike signs) and $\overline{F}_u$ (like signs) to control output emitter followers $\overline{F}_p$ and $\overline{F}_n$.

If C_n is high at P_0 indicating that $M_n > Y_{o2}$ numerically, and $\overline{F}_u$ is also high indicating like signs, then $C_n \overline{F}_u$ makes $\overline{F}_p$ high at P_0 to indicate a negative error which is used to reduce the shaft value of the converter. If F_u is high (unlike signs), then C_n F_u makes $\overline{F}_n$ high at P_0 to indicate a positive error which is used to increase the shaft value of the converter. Similar type logic, as shown in part b of figure 4-12, accounts for the cases in which $\overline{C}_n$ is high.

In the event that $M_n = Y_{o2}$, F_u will remain low throughout the word time, while $\overline{F}_u$ will be high. Flip-flops C_c and C_n will remain reset. $\overline{F}_o$ will be low at P_0 time to indicate no-error, since both F_u and C_c are low; $\overline{C}_n$ and $\overline{F}_u$ both high will make $\overline{F}_n$ high to indicate a positive error; but high C_c will make $\overline{F}_p$ high to indicate a positive error. The negative and positive error indications cancel each other.

Truth table 4-2 indicates all possible configurations of the error determining circuits at P_0 time.

Table 4-2

Error Determining Truth Table

F_uP0	0	0	1	1	0	1	1	0
C_NP_0	0	0	1	0	1	0	1	1
C_cP_0	0	1	1	1	1	0	0	0
	Zero $M_N = M_{ox} + Y_{o2}$	Pos. Error $M_N < M_{ox} + Y_{o2}$		Neg. Error $M_N > M_{ox} + Y_{o2}$			Never Occurs	
$\overline{F}_p$	1	0		1				
$\overline{F}_o$	0	1		1				
$\overline{F}_N$	1	1		0				
		Turn Disc C.C.W. As viewed from Brush Block end of Converter		Turn Disc C.W. As viewed from Brush Block end of Converter				

4-44. STARTING SEQUENCE

The computer is turned on and off through the agency of associated input-output equipment which is generally referred to in this manual as external control. When the computer is turned on (or off), power is sequentially applied (or removed) and definite patterns of logic control imposed. During normal computer operation the computer may be internally controlled by means of its program to cease computations and enter an idling phase. The idling phase may also be automatically terminated and computations resumed at predetermined time intervals. External and internal control methods are individually described in the following paragraphs.

4-45. External Control

Primary control of the starting sequence is effected by an OFF/STANDBY/RUN switch associated with external control. When this switch is set from OFF to STANDBY, power is sequentially applied to the computer, as shown in figure 4-13, to energize the drum motor (about 40 seconds for full speed) and all circuits except certain write amplifiers.

Signal R_3 is set high in external control to set phase flip-flops K_1, K_2, and K_3 (stop phase), to reset order register flip-flops 0_2 and 0_3 (Ad or At order), and to enter all "1's" into the instruction register I_0.

When the external control switch is set from STANDBY to RUN, signal R_4 is set high; then R_1 is set high and R_3 reset. The high R_4 signal ensures that the computer will not go from IR to SP phase by resetting K_1 with the term $K_2\overline{K}_3JR_4P_0$. The high R_1 signal gates the solitary S_{ot} pulse occurring at P_{18} of word 1 (this is the only P_{18} pulse on S_{ot}) to reset flip-flop K_1 ($k_1 = S_{ot}R_1\overline{T}_5T_4\overline{T}_2T_1$) which puts the computer in the wait beta phase. With the computer out of the stop phase, driver No. 4 signal goes low causing the reset of R_1. The reset of R_3 ($\overline{R}_3$ high) causes application of power (+ 35V DC) to the write amplifiers listed in the previous paragraph. Note that the pulse time indicator is synchronized with drum time previous to the setting of R_1 (refer to paragraph 4-27 for details).

During the wait beta phase a search is made for the fictitious operand (address: track 11111, sector 111111). When sector coincidence occurs, the computer enters the first word phase in which the fictitious operand is routed into the accumulator under control of the Ad or At order. From the first word phase the computer enters the wait alpha phase and a search is made for the instruction, the address of which is identical to that of the operand noted above. This address must contain the first programmed instruction, and must contain an order which will clear the accumulator of its meaningless data. Thereafter the computer operates normally under program control.

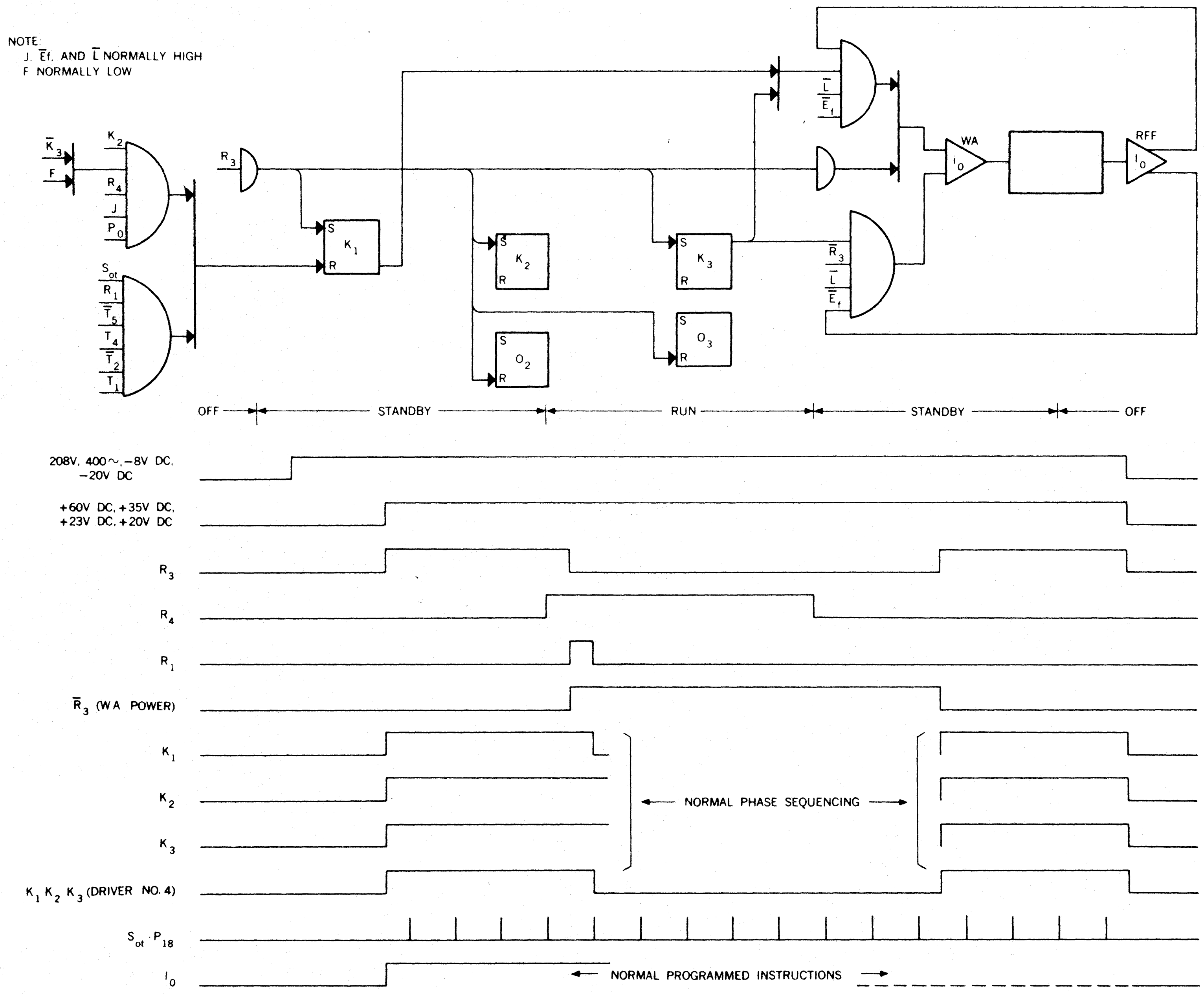


Figure 4-13. External Control, Logic and Timing Diagram

When the external control switch is set from RUN to STANDBY, R_4 goes low. The low state of R_4 causes the computer to go to stop at the end of the next instruction read-in phase. The stop phase signal ($K_1K_2K_3$) sends a high driver No. 4 signal to external control which causes power to be removed in the inverse sequence from which it was applied.

4-46. Internal Control

Under internal or program control the computer may be instructed to enter an idling period, ceasing all normal computation. In the idling period all power is removed except for idling power which is used to maintain flip-flops V_1 through V_6, C_4, C_5, T_s, and associated circuits in operation. When entering the idling phase the above noted flip-flops are logically controlled to form a shift register permitting a computed or predetermined number from the accumulator to be shifted into and stored in these flip-flops. After the number is stored, the flip-flops are altered through logical control to form a count-down register. The output from a real time generator in external control is used as input to the count-down register. Each input pulse reduces the binary count of the register by one. When the register reaches its next to the last count (000000001), the output configuration is decoded to generate a signal to initiate the starting sequence. On the last count (000000000) a second signal is decoded which generates another signal. This second signal continues the starting sequence to return the computer to normal operation.

The duration of the idling period is a function of the real time generator's frequency and the number transferred to the count-down register from the accumulator. This number may, of course, be either predetermined or computed, and may be varied for idling periods of different desired durations.

Two modified store orders and a special code to set an externally located H flip-flop are required to idle the computer and initiate count-down. It is assumed in the following description that the desired number to be counted down is in the accumulator and that the H flip-flop is set. The logic pertinent to the operation is shown in figure 4-14; a timing diagram appears in figure 4-15. The following paragraphs describe events in a phase by phase sequence.

4-47. Instruction Read-In (I_R) $K_1\ K_2\ \overline{K}_3$. Instruction is read into instruction register I_o and order (111) is shifted into order register $0_1 0_2 0_3$. At P_0 time, K_3 is set and K_1 is reset.

$$k_3 = K_1\ K_2\ P_0$$
$$\overline{k}_1 = K_2\ \overline{K}_3\ R_4\ J\ P_0$$

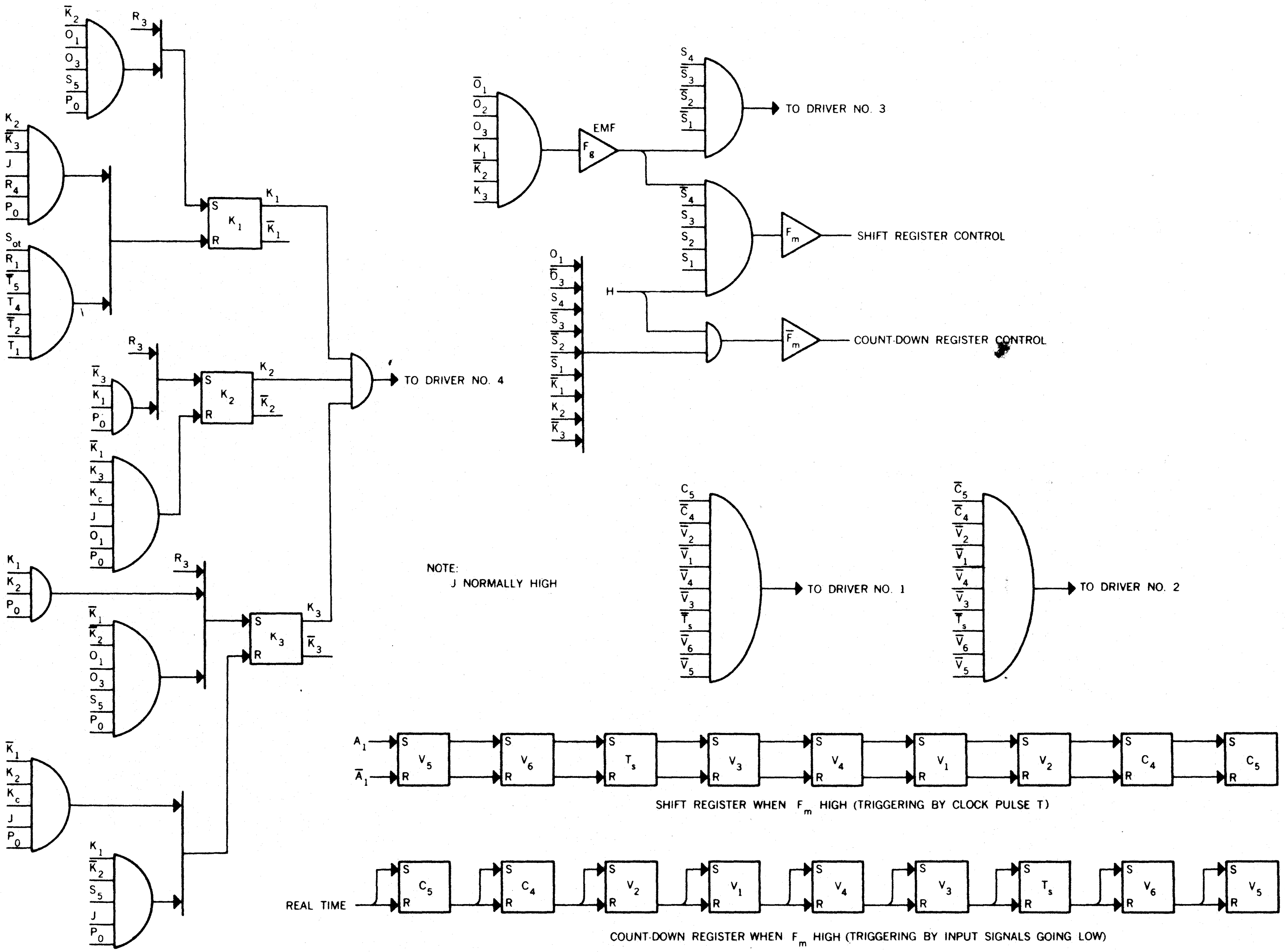
4-48. Wait Beta ($W\beta$) $\overline{K}_1\ K_2\ K_3$. Track beta ($T\beta$) portion of instruction is shifted into track selection register by high F_s gating signal. The code is $S_5\ \overline{S}_4\ S_3\ S_2\ S_1$. When sector coincidence occurs, K_c is high at P_0 to reset K_2 and K_3.

$$f_s = \overline{K}_1\ K_2\ K_3\ \overline{T}_5\ (\overline{T}_3 + \overline{T}_2)$$
$$\overline{k}_2 = \overline{K}_1\ K_3\ K_c\ J\ 0_1\ P_0$$
$$\overline{k}_3 = \overline{K}_1\ K_2\ K_c\ J\ P_0$$

4-49. First Word (FW) $\overline{K}_1\ \overline{K}_2$. At P_0, flip-flop 0_1 of the order register is reset by S_5 and K_1 and K_3 are set.

$$\overline{o}_1 = \overline{K}_1\ \overline{K}_2\ 0_2\ 0_3\ S_5\ P_0$$
$$k_1 = K_2\ 0_1\ 0_3\ S_5\ P_0$$
$$k_3 = \overline{K}_1\ \overline{K}_2\ 0_1\ 0_3\ S_5\ P_0$$

4-50. Additional Words (AW) $K_1\ \overline{K}_2\ K_3$. The order register contains the conditional transfer order $\overline{0}_1\ 0_2\ 0_3$. The AW-Tc code ($K_1\ \overline{K}_2\ K_3\ \overline{0}_1\ 0_3$) causes emitter follower F_g to become high throughout the entire phase. The high F_g signal in conjunction with the high H signal and the code $\overline{S}_4\ S_3\ S_2\ S_1$ makes emitter follower F_m high. The high F_m signal forms flip-flops V_1 through V_6, C_4, C_5, and T_s into a shift register and gates the contents of the accumulator into this register. At P_0 of the present word sector coincidence occurs, and K_3 is reset.


$\overline{K}_2$ O_1 O_3 S_5 $\overline{P}_0$
R_3
K_2 $\overline{K}_3$ J R_4 P_0
S_{ot} R_1 $\overline{T}_5$ T_4 $\overline{T}_2$ T_1
S R K_1 K_1 $\overline{K}_1$
R_3
$\overline{K}_3$ K_1 P_0
S R K_2 K_2 $\overline{K}_2$
$\overline{K}_1$ K_3 K_c J O_1 P_0
TO DRIVER NO. 4
K_1 K_2 P_0
R_3
$\overline{K}_1$ $\overline{K}_2$ O_1 $\overline{O}_3$ S_5 P_0
S R K_3 K_3 $\overline{K}_3$
$\overline{K}_1$ K_2 K_c J P_0
K_1 $\overline{K}_2$ S_5 J P_0
$\overline{O}_1$ O_2 O_3 K_1 $\overline{K}_2$ K_3
EMF
F_g
S_4 $\overline{S}_3$ $\overline{S}_2$ $\overline{S}_1$
TO DRIVER NO. 3
$\overline{S}_4$ S_3 S_2 S_1
F_m
SHIFT REGISTER CONTROL
O_1 $\overline{O}_3$ S_4 $\overline{S}_3$ $\overline{S}_2$ $\overline{S}_1$ $\overline{K}_1$ K_2 $\overline{K}_3$
H
$\overline{F}_m$
COUNT-DOWN REGISTER CONTROL
C_5 $\overline{C}_4$ $\overline{V}_2$ $\overline{V}_1$ $\overline{V}_4$ $\overline{V}_3$ $\overline{T}_s$ $\overline{V}_6$ $\overline{V}_5$
TO DRIVER NO. 1
$\overline{C}_5$ $\overline{C}_4$ $\overline{V}_2$ $\overline{V}_1$ $\overline{V}_4$ $\overline{V}_3$ $\overline{T}_s$ $\overline{V}_6$ $\overline{V}_5$
TO DRIVER NO. 2
A_1 $\overline{A}_1$
S R V_5 S R V_6 S R T_s S R V_3 S R V_4 S R V_1 S R V_2 S R C_4 S R C_5
REAL TIME
S R C_5 S R C_4 S R V_2 S R V_1 S R V_4 S R V_3 S R T_s S R V_6 S R V_5

NOTE:
J NORMALLY HIGH

SHIFT REGISTER WHEN F_m HIGH (TRIGGERING BY CLOCK PULSE T)

COUNT-DOWN REGISTER WHEN F_m HIGH (TRIGGERING BY INPUT SIGNALS GOING LOW)

Figure 4-14. Internal Idling Control, Logic Diagram

$f_g = K_1 \overline{K}_2 K_3 \overline{0}_1 0_2 0_3$

$fm = F_g H \overline{S}_4 S_3 S_2 S_1$

$\overline{k}_3 = K_1 \overline{K}_2 S_5 J P_0$ (Flip-flop S_5 is reset during P_8 through P_3 if sector coincidence does not occur.)

4-51. Last Word (LW) $K_1 \overline{K}_2 \overline{K}_3$. During P_{24} through P_{20} ($T\alpha$) the track address of the next instruction is shifted into the track selection register. Emitter follower $\overline{F}_m$ is made high to form the V_1 through V_6, C_4, C_5, and T_s flip-flops into a count-down register. These flip-flops remain as a count-down register until the idling period is completed. Actual count-down begins with the first real-time pulse received. At P_0 time, K_2 is set.

$$f_s = K_1 \overline{K}_2 \overline{K}_3 \overline{T}_5 \overline{T}_4 (\overline{T}_3 + \overline{T}_2)$$
$$k_2 = K_1 \overline{K}_3 P_0$$

4-52. Instruction Read-In (IR) $\overline{K}_1 \overline{K}_2 K_3$. The next instruction is read into the instruction register I_o. The sector address of the instruction is determined by the timing of the program since there is no wait alpha phase. The order, which is another store order, is shifted into the order register. K_1 is reset and K_3 set at P_0.

4-53. Wait Beta (Wβ) $\overline{K}_1 K_2 K_3$. The modified store order code is shifted into the track selection register by the high F_s gating signal. The code is $S_5 S_4 \overline{S}_3 \overline{S}_2 \overline{S}_1$. When sector coincidence occurs, K_c is high at P_0 to reset K_2 and K_3.

$$f_s = \overline{K}_1 K_2 K_3 \overline{T}_5 (\overline{T}_3 + \overline{T}_2)$$
$$k_2 = \overline{K}_1 K_3 K_c J 0_1 P_0$$
$$k_3 = \overline{K}_1 K_2 K_c J P_0$$

4-54. First Word (FW) $\overline{K}_1 \overline{K}_2$. At P_0 flip-flop 0_1 of the order register is reset by S_5, and K_1 and K_3 are set.

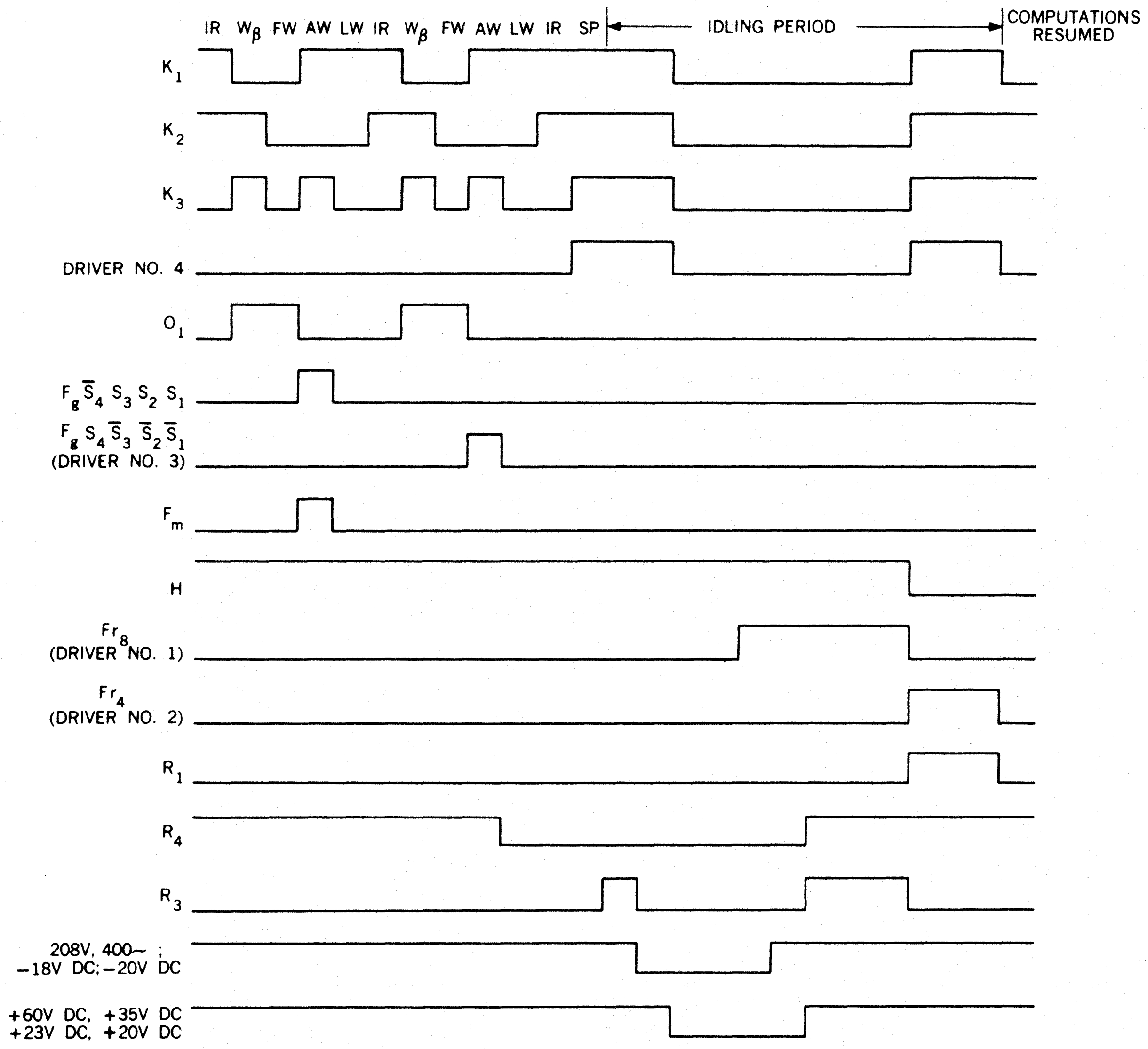


Figure 4-15. Internal Idling Control, Timing Diagram

$$\overline{o}_1 = \overline{K}_1\ \overline{K}_2\ 0_2\ 0_3\ S_5\ P_0$$
$$k_1 = \overline{K}_2\ 0_1\ 0_3\ S_5\ P_0$$
$$k_3 = \overline{K}_1\ \overline{K}_2\ 0_1\ 0_3\ S_5\ P_0$$

4-55. Additional Words (AW) $K_1\ \overline{K}_2\ K_3$. The order register contains the conditional transfer order $\overline{0}_1\ 0_2\ 0_3$. The AW-Tc code $(K_1\ \overline{K}_2\ K_3\ \overline{0}_1\ 0_3)$ causes emitter follower F_g to become high for the entire phase. The high F_g signal in conjunction with the high H signal and the code $S_4\ \overline{S}_3\ \overline{S}_2\ \overline{S}_1$ generates a high driver No. 3 signal to external control to make R_4 low. At P_0 of the present word, sector coincidence is programmed to occur, and K_3 is reset.

$$f_g = K_1\ \overline{K}_2\ K_3\ \overline{0}_1\ 0_2\ 0_3$$
$$k_3 = K_1\ \overline{K}_2\ S_5\ J\ P_0$$

4-56. Last Word (LW) $K_1\ \overline{K}_2\ \overline{K}_3$. During P_{24} through P_{20} $(T\alpha)$ the track address of the next instruction is shifted into the track selection register by a high F_s signal. At P_0 time K_2 is set.

$$f_s = K_1\ \overline{K}_2\ \overline{K}_3\ \overline{T}_5\ \overline{T}_4\ (\overline{T}_3 + \overline{T}_2)$$
$$k_2 = K_1\ \overline{K}_3\ P_0$$

4-57. Instruction Read-In (IR) $K_1\ K_2\ \overline{K}_3$. At P_0 time K_3 is set, but the low R_4 signal from external control inhibits the resetting of K_1. The computer therefore enters the stop phase $(K_1\ K_2\ K_3)$.

$$k_3 = K_1\ K_2\ P_0$$
$$\overline{k}_1 = K_2\ \overline{K}_3\ J\ R_4\ P_0\ (R_4 \text{ is low at this time})$$

4-58. Stop (SP) $K_1\ K_2\ K_3$. The stop phase code generates a high driver No. 4 signal to external control where R_3 is set high to remove write amplifier power, then drum and circuit power. Idling power remains on.

4-59. Idling Period and Idle to Run. The idling period continues until the count-down register reaches the next to the last count (000000001), at which time F_{r8} (driver No. 1) is made high, which in turn causes the power sequence to start, and sets R_3 and R_4 in external control. The high R_3 signal sets flip-flops K_1 K_2 K_3 0_2 0_3 and causes "1's" to be written in the instruction register as described in the description of external control. When the count-down register reaches the last count (000000000), F_{r4} (driver No. 2) is made high which causes the reset of the H flip-flop, the reset of R_3 which applies write amplifier power, and the setting of R_1 which yields the start signal. The occurrence of the P_{18} pulse on S_{ot} resets R_1 and begins normal computations as described under external control.

$$f_{r8} = \overline{V}_6 \; \overline{V}_5 \; \overline{T}_s \; \overline{V}_4 \; \overline{V}_3 \; \overline{V}_1 \; \overline{V}_2 \; \overline{C}_4 \; \overline{C}_5$$
$$f_{r4} = \overline{V}_6 \; \overline{V}_5 \; \overline{T}_s \; \overline{V}_4 \; \overline{V}_3 \; \overline{V}_1 \; \overline{V}_2 \; \overline{C}_4 \; \overline{C}_5$$

4-60. CLEAR AND ADD ORDER (At) $\overline{0}_1$ $\overline{0}_2$ $\overline{0}_3$

The clear and add order routes the selected operand word from main memory to the accumulator register while destroying the previous contents of the accumulator.

The clear and add instruction is located during the wait alpha Wα phase, $\overline{K}_1$ K_2 $\overline{K}_3$. It is routed from main memory into the instruction register with the order being stored in the order register during the instruction read-in IR phase, K_1 K_2 $\overline{K}_3$. During the wait beta Wβ phase, $\overline{K}_1$ K_2 K_3, the operand is located in main memory and the K_c flip-flop is reset at P_0 by the high $\overline{0}_3$ of the order register.

During the first word FW phase, $\overline{K}_1$ $\overline{K}_2$ $\overline{K}_3$, flip-flops C_a and A_1 are reset at P_{24} time, see figure 4-16, and remain reset for the entire word time. Therefore, the "anded" expression $\overline{A}_1$ $\overline{C}_A$ F_g maintains a high emitter follower $\overline{F}_a$ output which gates the outputs of flip-flop D_1 to write amplifier a_0. Signals $\overline{E}_f$ and F_b are normally high. On the AND gates controlling write amplifier a_0, therefore, the D_1 outputs are the only varying signals and have full control of the gates. Since D_1 simply copies the word from M_{oy}, a_0, in turn, simply copies D_1. At the end of one word time the operation is complete, and K_2 is

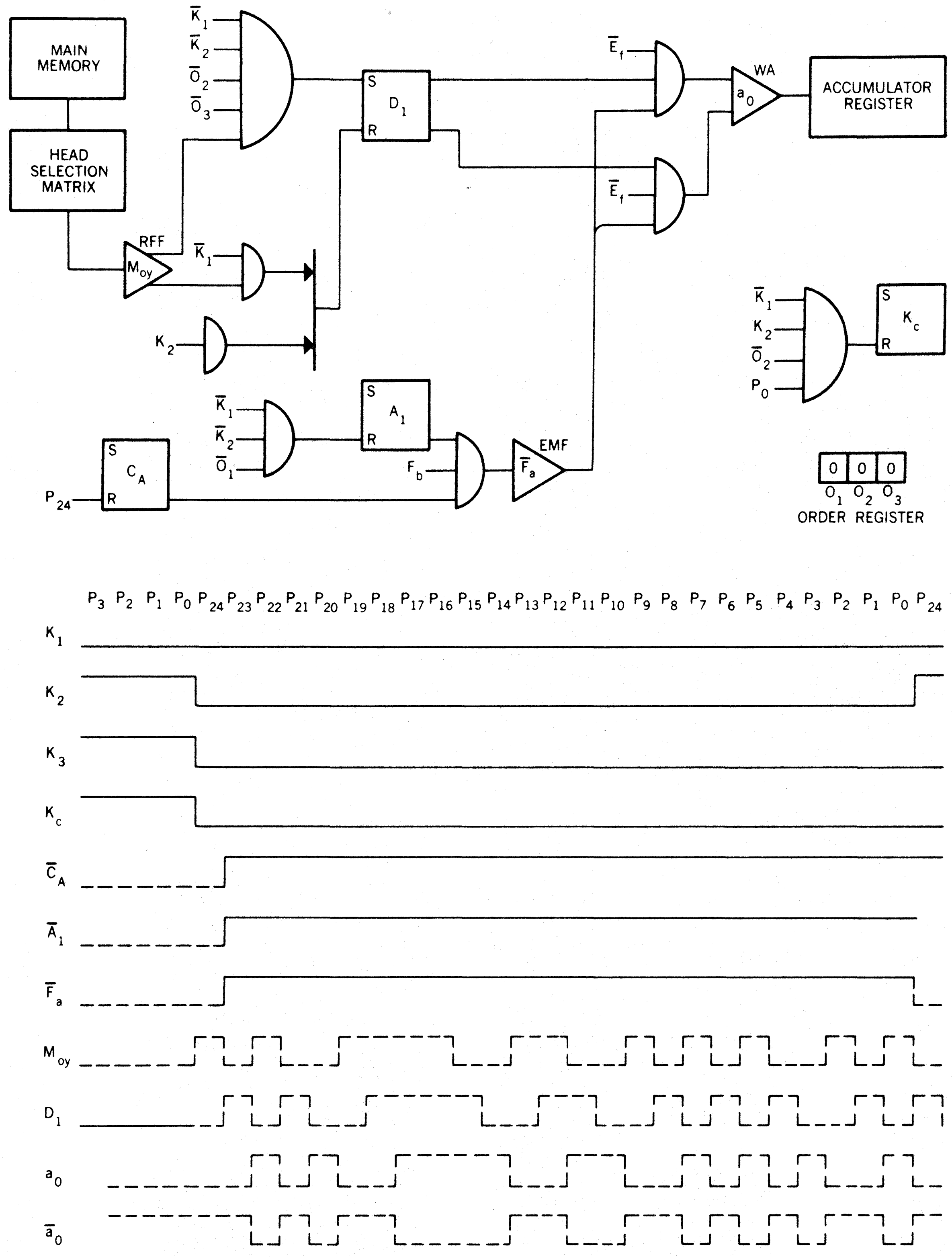


Figure 4-16. Clear and Add Order, Logic and Timing Diagram

set to put the computer in the wait alpha phase with the new word normally recirculating in the accumulator register.

4-61. ADD ORDER (Ad) $0_1 \overline{0}_2 \overline{0}_3$

In the add order, figure 4-17, a sum is formed of the number contained in the accumulator (A_0) and the operand from main memory (M_{oy}). The contents of the accumulator is added to the number specified by the β address of the instruction. Addition is begun and completed in one word time during the FW phase. The final sum is stored in the A_0 register.

The logic for the serial adder is the same for both addition and subtraction with the exception of the input logic carry flip-flop (C_a). The output of the carry control flip-flop, K_c, determines the operation of C_a. When $K_c = 0$, C_a operates with carry logic and the add operation takes place. Conversely, when $K_c = 1$, C_a operates with borrow logic and a subtraction operation takes place. The K_c flip-flop assumes its proper state at P_{24} of the FW phase.

In the FW phase A_1 and D_1 copy the A_0 and M_{oy} registers respectively. Due to the one bit delay of the A_1 and D_1 flip-flops, the L.S.D. information which appears at A_0 and M_{oy} at P_{24} time will appear in the adder and the L.S.D. of the sum will be written on A_0 at P_{23} time.

The logic for the add order is:

$$\begin{aligned}
a_1 &= A_0 \overline{K}_1 \, 0_1 \, F_j \, G \\
\overline{a}_1 &= \overline{A}_0 \overline{K}_1 \, 0_1 \, F_j \, G \\
d_1 &= \overline{K}_1 \overline{K}_2 M_{oy} \overline{0}_2 \overline{0}_3 \\
\overline{d}_1 &= \overline{K}_1 \overline{M}_{oy} \\
c_a &= \overline{K}_c A_1 D_1 \overline{P}_{24} \\
\overline{c}_a &= \overline{K}_c \overline{A}_1 \overline{D}_1 + P_{24}
\end{aligned}$$

The signal F_j is an emitter follower amplifier that is high for all orders except certain modified store orders, and G is a test signal that is high during normal operation.

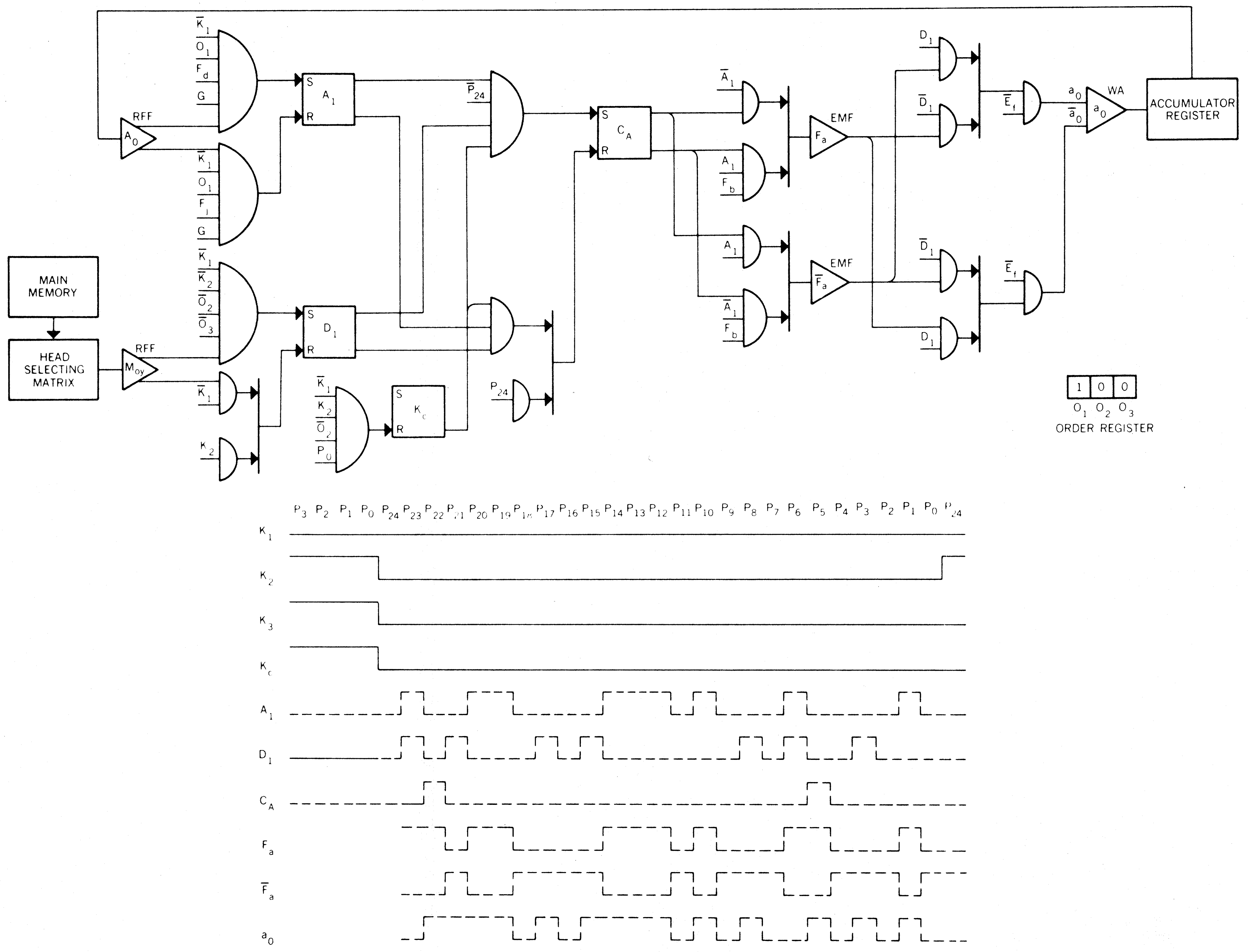


Figure 4-17. Add Order, Logic and Timing Diagram

The second part of the add logic is centered around the F_a and $\overline{F}_a$ emitter followers. The input logic for these emitter followers is:

$$f_a = \overline{A}_1 C_a + A_1 \overline{C}_a F_b$$
$$\overline{f}_a = A_1 C_a + \overline{A}_1 \overline{C}_a F_b$$

The F_b signal is an emitter follower output that is high at all times except for Su AW and Sr AW.

The final step in the add order involves writing the sum of the accumulator contents and the operand from the main memory on the accumulator register. This is accomplished by the A_0 write amplifier. Input logic for the write amplifier is:

$$a_0 = (F_a \overline{D}_1 + \overline{F}_a D_1) \overline{E}_f$$
$$\overline{a}_0 = (F_a D_1 + \overline{F}_a \overline{D}_1) \overline{E}_f$$

$\overline{E}_f$ is a test signal which is high during normal operation.

4-62. SUBTRACT ORDER (Su) 0_1 0_2 $\overline{0}_3$.

The subtract order (Su) allows the computer to subtract the contents of the accumulator (A_0) from the contents of the main memory (M_{oy}). This type of subtraction (M_{oy} - A_0) is defined as abnormal subtraction (refer to paragraph 4-40) and is accomplished during the FW phase.

The subtraction logical diagram and waveforms are illustrated in figure 4-18. The logic for Su is the same as the logic for the add order (Ad) with the exception of the carry logic. For the Su order the C_a flip-flop operates with borrow logic. The input logic for the C_a flip-flop is:

$$c_a = \overline{K}_1 \overline{K}_2 K_c A_1 \overline{D}_1 0_1 \overline{P}_{24}$$
$$\overline{c}_a = P_{24} + \overline{K}_1 K_2 0_1 K_c \overline{A}_1 D_1$$

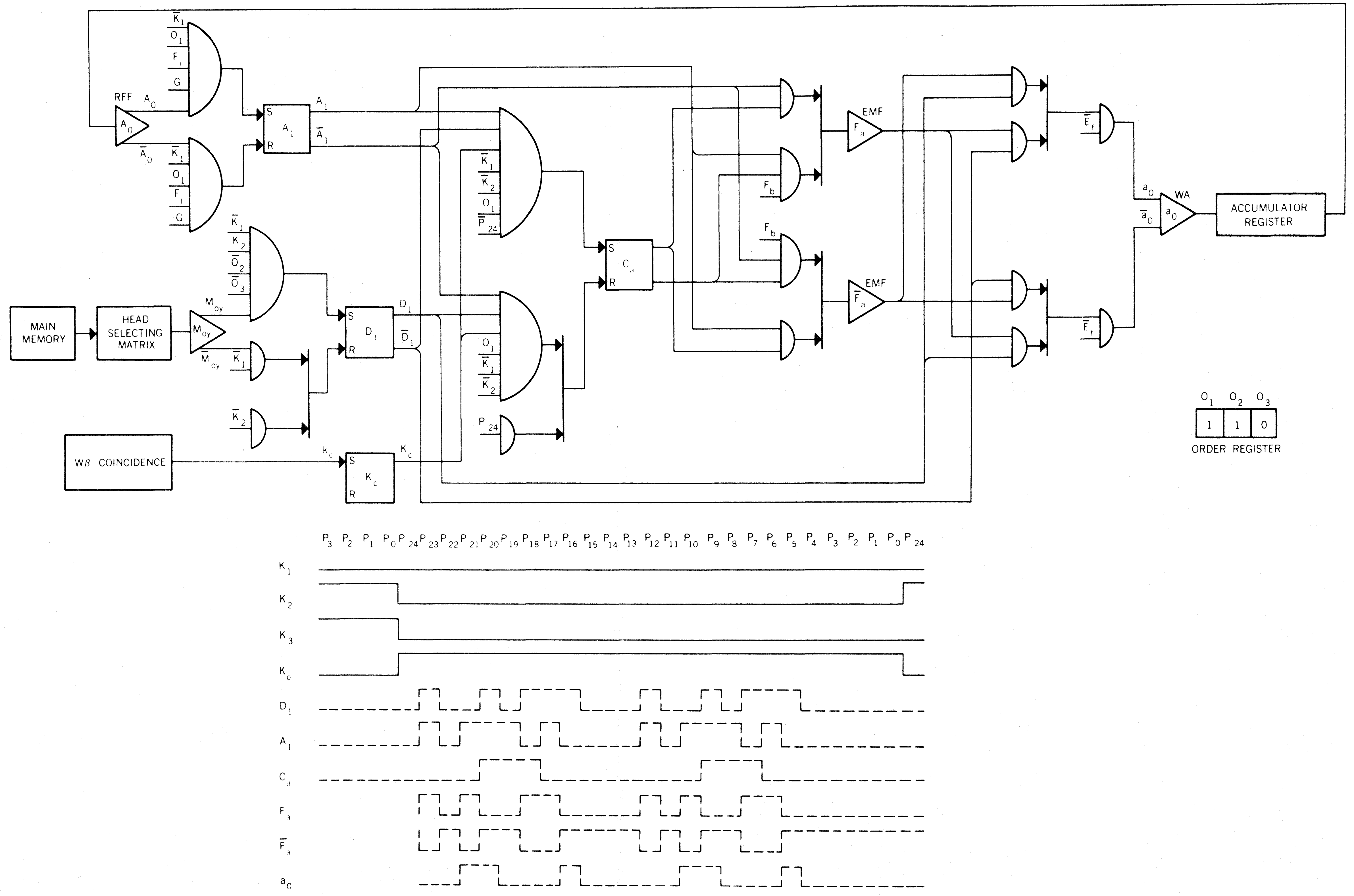


Figure 4-18. Subtract Order, Logic and Timing Diagram

4-63. EXTRACT ORDER (Ex) $\overline{0}_1 \overline{0}_2 0_3$.

The extract order is frequently used to modify instructions by changing the address portion of the instruction. The Ex order allows the computer to take the logical or boolean product of the contents of the accumulator (A_0) and the contents of the main memory (M_{oy}). This operation results in writing a "1" in the accumulator wherever there is a "1" in both A_0 and M_{oy}, and "0's" everywhere else. By this method data contained in a given part of the word can be selected by a control word containing all "1's" in that part of the word. This number can be either in the accumulator or main memory as the numbers are treated identically.

Figure 4-19 is the logical diagram and waveforms of the extract order. Logic for the extract order is as follows:

$$\begin{aligned} \overline{a}_1 &= \overline{K}_1 \overline{K}_2 \overline{0}_1 \\ d_1 &= \overline{K}_1 \overline{K}_2 M_{oy} (A_0 \overline{0}_1 0_3) \\ \overline{d}_1 &= \overline{K}_1 \overline{M}_{oy} + \overline{A}_0 0_3 \overline{K}_1 \\ c_a &= \overline{K}_c A_1 D_1 \overline{P}_{24} \end{aligned}$$

The extract operation centers around the D_1 flip-flop, as the A_1 flip-flop is reset for FW Ex. Whenever A_0 and M_{oy} are both "1's" during FW Ex, the D_1 flip-flop is set. The D_1 flip-flop is reset when a "0" occurs in either A_0 or M_{oy}. The carry flip-flop operates with add logic since K_c is low during FW Ex, An example of the extract operation is shown below:

$$\begin{array}{ll} A_0 & 0\;1\;0\;1\;1\;1\;0\;1\;1\;0\;1\;1\;1 \\ M_{oy} & \underline{0\;0\;0\;0\;0\;1\;1\;1\;1\;0\;0\;0\;0} \\ D_1 & 0\;0\;0\;0\;0\;1\;0\;1\;1\;0\;0\;0\;0 \end{array}$$

4-64. CONDITIONAL TRANSFER ORDER (Tc) $\overline{0}_1 0_2 0_3$.

The Tc operation selects one of two addresses for the next instruction based upon the sign digit of the accumulator. If the sign digit is nega-

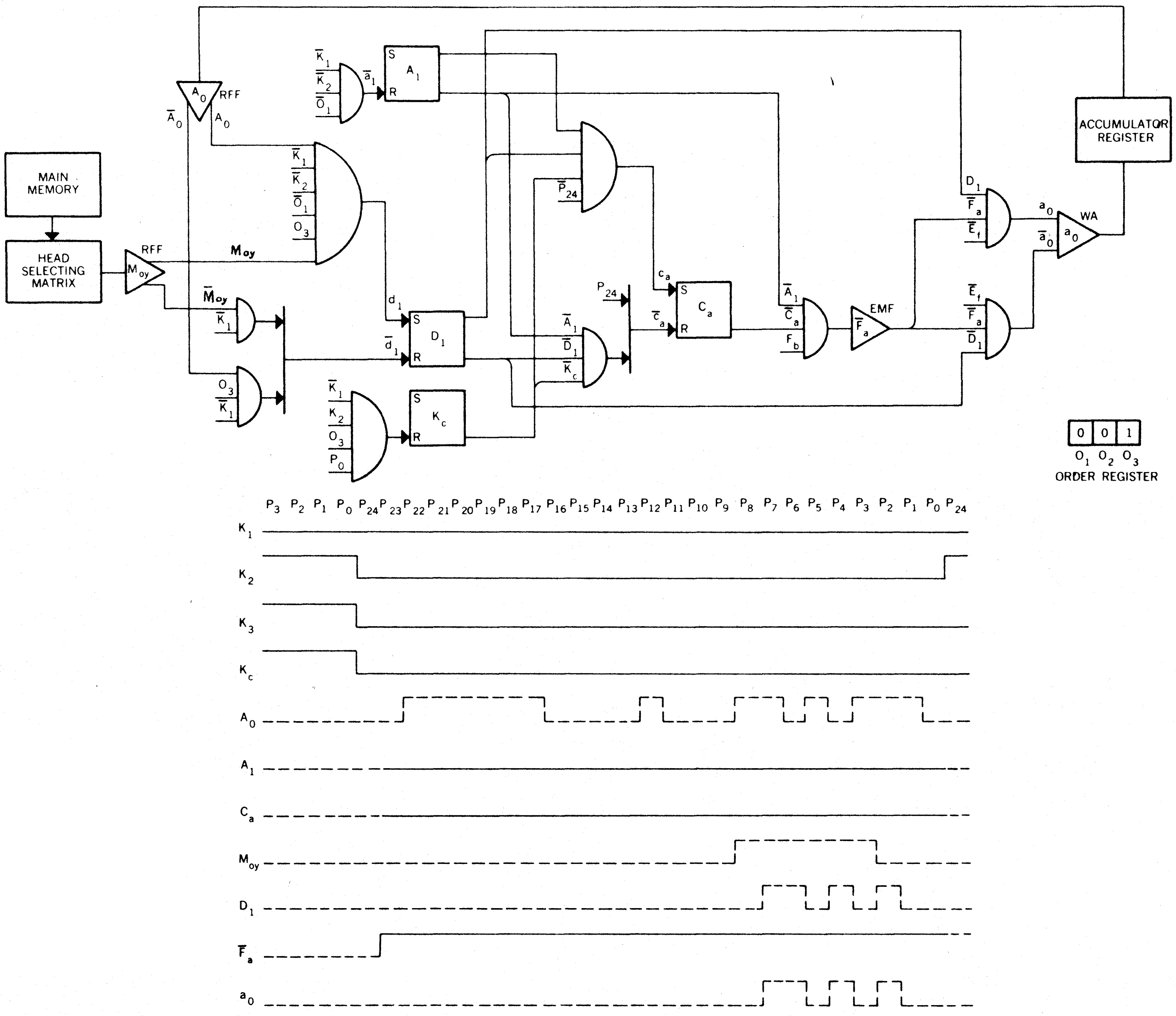


Figure 4-19. Extract Order, Logic and Timing Diagram

tive (high), the location of the next instruction is taken from the α address of the present instruction. If the sign digit is positive (low), the next instruction is taken from the β address of the present instruction. The sign digit occurs in A_1 at P_{24} time. The contents of the accumulator remains unchanged during the Tc order. The Tc order is accomplished during the Wβ phase.

When the sign digit is negative, K_3 is reset at P_{23} time of the first word and the computer goes to the Wα phase. The K_3 flip-flop is reset in this instance by the following logic:

$$\overline{k}_3 = A_1 \, P_{24} \, \overline{K}_1 \, \overline{0}_1 \, 0_2 \, 0_3$$

When the sign digit is positive, the computer goes into the IR phase. To accomplish this, K_1 is set and K_3 is reset by the following logic at the end of one word time:

$$k_1 = K_2 \, K_c \, P_0 \, \overline{0}_1 \, 0_2 \, 0_3$$
$$k_3 = \overline{K}_1 \, K_2 \, K_c \, J \, P_0$$

The Tc order code also occurs during AW and LW phases when performing modified store orders; however, its function is entirely divorced from the normal conditional transfer order (Tc Wβ) described above. The conditions for Tc occurring in the AW phase are 15 relay set commands of the modified store order wherein the information stored in S_5 is shifted to the 0_1 flip-flop of the order register by the following logic:

$$\overline{o}_1 = 0_2 \, 0_3 \, \overline{K}_1 \, \overline{K}_2 \, S_5 \, P_0$$

This type of modified store order occurs at P_0 of FW where the K_1 and K_3 flip-flops are set by the following logic and the computer goes to the AW phase:

$$k_1 = \overline{K}_2 \, P_0 \, 0_1 \, 0_3 \, S_5$$
$$k_3 = \overline{K}_1 \, \overline{K}_2 \, P_0 \, 0_1 \, 0_3 \, S_5$$

4-65. NORMAL STORE ORDERS (Sr) 0_1 0_2 0_3 $\overline{S}_5$ $\overline{S}_4$.

The normal store orders provide capabilities in the program of storing information from the accumulator register A_0 to one or more of the temporary storage tracks M_{ov}, M_{ox}, and M_{oy} or the sigmator track Y_o . The normal store order is a one word operation which is completed in the FW phase. The store order code (0_1 0_2 0_3) in conjunction with $\overline{S}_5$ and $\overline{S}_4$ of the track selection register defines normal store orders. Specific configurations of flip-flops S_3, S_2, and S_1 define particular storage locations. The codes are programmed and stored in the main memory as part of an instruction word.

The order code is shifted into the order register during the IR phase with flip-flops 0_1, 0_2, and 0_3 being set. During the Wβ phase the Tβ portion (P_{19} through P_{15}) of the instruction register I_0 is shifted into the track selection register. At sector coincidence (K_c P_0) the computer enters FW. During FW the information from the accumulator register A_0 is accepted by the temporary storage or sigmator track write amplifiers as determined by the code stored in the track selection register flip-flops S_1, S_2, and S_3 as qualified by the term 0_2 0_3 $\overline{S}_4$ $\overline{S}_5$. $\overline{K}_1$ $\overline{K}_2$. The FW phase is terminated by setting flip-flop K_2 by the term K_1 0_1 0_2 $\overline{S}_5$ $\overline{S}_4$ P_0.

The operations performed by the normal store order are:

Number Code								
0_1	0_2	0_3	S_1	S_2	S_3	S_4	S_5	
1	1	1	0	0	0	0	0	A_0 stored in Y_0
1	1	1	1	0	0	0	0	A_0 stored in M_{ov}
1	1	1	0	1	0	0	0	A_0 stored in M_{ox}
1	1	1	1	1	0	0	0	A_0 stored in M_{ov} and M_{ox}
1	1	1	0	0	1	0	0	A_0 stored in M_{oz}
1	1	1	1	0	1	0	0	A_0 stored in M_{ov} and M_{oz}
1	1	1	0	1	1	0	0	A_0 stored in M_{ox} and M_{oz}
1	1	1	1	1	1	0	0	A0 stored in M_{ov} and M_{ox} and M_{oz}

Logic

	Copy A_0	Storage Location		FW	Sr
mov =	A_0	$S_1 \bar{S}_4 \bar{S}_5$	Track 28	$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$
$\overline{\text{mov}}$ =	$\bar{A}_0$	$S_1 \bar{S}_4 \bar{S}_5$		$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$
mox =	A_0	$S_2 \bar{S}_4 \bar{S}_5$	Track 29	$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$
$\overline{\text{mox}}$ =	$\bar{A}_0$	$S_2 \bar{S}_4 S_5$		$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$
moz =	A_0	$S_3 \bar{S}_4 \bar{S}_5$	Track 30	$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$
$\overline{\text{moz}}$ =	$\bar{A}_0$	$S_3 \bar{S}_4 \bar{S}_5$		$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$
yo =	A_0	$\bar{S}_1\bar{S}_2\bar{S}_3\bar{S}_4\bar{S}_5$	write in yo	$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$
$\overline{\text{yo}}$ =	$\bar{A}_0$	$\bar{S}_1\bar{S}_2\bar{S}_3\bar{S}_4\bar{S}_5$		$\bar{K}_1 \bar{K}_2$	$0_2\ 0_3$

$$k_2 = \bar{K}_1\ P_0\ J\ \left[\bar{0}_1\ 0_3 + 0_1\ 0_2\ \bar{S}_5\ (\bar{S}_4 + \bar{S}_3)\right]$$

4-66. MODIFIED STORE ORDER $0_1\ 0_2\ 0_3\ \bar{S}_5\ S_4$

The modified store order provides the program with the capabilities of reading into A_0 from three sources: extracting and left or right shift, relay set orders and program stop, and left or right shift.

The modified store order can be either a one or additional word operation. The binary code of the order register and the combination of the track selector flip-flops provide the code for the command to be selected. The Tβ numbers are programmed and stored in the main memory as part of the instruction word.

The order code is shifted in the order register during the IR phase. The order register flip-flops are all high to define the store order, while the S_5 flip-flop is low and the S_4 flip-flop is high which determines that the store command is a modified store. During the Wβ phase, the track information (P_{19} through P_{15}) is shifted into the track selector flip-flops. This portion of the track determines the exact modified store command to be used. At coincidence, the computer enters the FW phase. If the command selected is the read-in command (Y_{o1} + Y_{o2} + M_n into A_0) the operation is completed in one word time.

During the extract (A_0 with M_n) and recirculate (A_0) commands, an additional word phase is required. A modified store code on k_1 and k_3 $\left[0_1\ 0_3\ (S_5 + S_4\ S_3)\right]$ causes the control transfer from FW to AW. The S_2 clarifier in the command determines whether a left or right shift modification occurs. That is, when S_2 is zero a right shift occurs, and when S_2 is one a left shift occurs. The program determines how many words are necessary, and one bit of the A_0 is shifted for each additional word. In the last word phase the contents of A_0 are recirculated for one word. This does not modify the contents of A_0.

Logic for FW read-in commands.

FW	S_r	Read-in	Y_{o1}	Y_{o2}	M_n
$a_1 = \bar{K}_1\ \bar{K}_2$	$0_2\ 0_3$	$\bar{S}_5\ S_4\ \bar{S}_3$	$(Y_{o1}\ S_2\ S_1$	$+\ Y_{o2}\ \bar{S}_2$	$+\ M_n\ \bar{S}_1)$
$\bar{a}_1 = \bar{K}_1\ \bar{K}_2$	$0_2\ 0_3$	$\bar{S}_5\ S_4$	$(\bar{S}_3\ \bar{Y}_{o1}\ S_2\ S_1$	$+\ \bar{Y}_{o2}\bar{S}_2\bar{S}_3$	$+\ \overline{M_n}\ \bar{S}_1)$

Number code

0_1	0_2	0_3	S_1	S_2	S_3	S_4	S_5	
1	1	1	1	0	0	1	0	Y_{o2} read into A_0
1	1	1	0	1	0	1	0	M_n read into A_0
1	1	1	1	1	0	1	0	Y_{o1} read into A_0

Logic for FW Extract (A_0 with M_n) and recirculate (A_0)

$a_1 = G\ A_0\ \bar{K}_1\ 0_1\ (F_j + S_3 M_n + S_1\ S_3)$

$\bar{a}_1 = G\ \bar{A}_0\ K_1\ 0_1\ (F_j + S_3) + \bar{K}_1\ \bar{K}_2\ 0_2\ 0_3\ \bar{S}_5\ S_4\ (\overline{M_n}\ \bar{S}_1)$

Number code

0_1	0_2	0_3	S_1	S_2	S_3	S_4	S_5		
1	1	1	0	0	1	1	0	Ex A_0 with M_n	Right Shift
1	1	1	1	0	1	1	0	Recirculate	

A one results when A_0 and M_n both contain a one, and a zero results when a zero occurs in A_0, S_3 is high, or when a zero occurs in M_n and the S_1 flip-flop is reset.

Example:

$$\begin{array}{ll} A_0 & 1\;0\;0\;1\;0\;1\;1\;1\;0\;1 \\ M_n & \underline{0\;0\;0\;1\;1\;1\;1\;1\;1\;0} \\ & 0\;0\;0\;1\;0\;1\;1\;1\;0\;1 \end{array}$$

The right shift modification occurs during the AW phase or the LW phase. For the right shift in AW, A_0 is copied at all times except P_{24}, or when F_b is low. The sign digit is extended to P_{24} by C_a, which is not reset until P_{24}.

Logic for AW right shift.

$$f_a = \overline{C}_A\ (A_0\ 0_1\ 0_2\ K_1\ \overline{K}_2\ K_3\ \overline{S}_2\ \overline{P}_{24})$$
$$\overline{f}_a = \overline{C}_A\ (\overline{A}_0\ 0_1\ 0_2\ K_1\ \overline{K}_2\ K_3\ \overline{S}_2\ \overline{P}_{24})$$

At coincidence ($S_5\ P_0$) the K_3 flip-flop is reset and the LW phase is entered.

Logic for LW right shift.

$$f_a = \overline{C}_A\ (A_1\ F_b)$$
$$\overline{f}_a = \overline{C}_A\ (\overline{A}_1\ F_b)$$
$$f_b = \overline{0}_1 + \overline{0}_2 + \overline{K}_1 + K_2 + \overline{K}_3 + S_2 + P_{24}$$

During the right shift command in the last word phase the contents of A_0 are recirculated for one word by the following logic:

$$a_1 = G\ A_0\ (K_2 + K_1\ 0_2)$$
$$\overline{a}_1 = G\ \overline{A}_0\ (K_2 + K_1\ 0_2)$$

Example of right shift (scope view)

1 1 0 1 1 1 1 1 0 1 . 1
1 1 0 1 1 1 1 1 0 1 1 . 1 Right-shift

The FW extract (A_0 with M_n) and recirculate (A_0) logic is identical for either left or right shift. The number code for this operation is the same in both cases with the exception of the S_2 clarifier. For the left shift modification, S_2 contains a one. The left shift also occurs during the AW and LW phases. Normal add and carry logic is used for the left shift command.

Logic for the left shift modification:

$$a_1 = G\, A_0\, (K_2 + K_1\, 0_2)$$
$$\overline{a}_1 = G\, \overline{A}_0\, (K_2 + K_1\, 0_2)$$
$$d_1 = K_1\, \overline{K}_2\, K_3\, 0_1\, 0_2\, 0_3\, S_2\, A_0$$

For a left shift during the LW phase, A_0 is recirculated for one word. The D_1 flip-flop is reset in the LW phase by the following logic:

$$\overline{d}_1 = \overline{A}_0\, 0_3\, (0_2 + \overline{K}_1)$$

The remaining 16 modified store orders include the program stop and 15 relay set commands. The number code for the track selector flip-flops during the program stop and relay set commands is listed below.

S_5	S_4	S_3	S_2	S_1	
1	0	0	0	0	Stop
1	0	0	0	1	Relay Set
1	0	0	1	0	Relay Set
1	0	0	1	1	Relay Set
1	0	1	0	0	Relay Set
1	0	1	0	1	Relay Set
1	0	1	1	0	Relay Set
1	0	1	1	1	Relay Set
1	1	0	0	0	Relay Set
1	1	0	0	1	Relay Set
1	1	0	1	0	Relay Set
1	1	0	1	1	Relay Set
1	1	1	0	0	Relay Set
1	1	1	0	1	Relay Set
1	1	1	1	0	Relay Set
1	1	1	1	1	Relay Set

These commands are recirculated during the FW phase then shifted to Tc AW. The 15 relay set commands vary with computer application, so no attempt at logic description is included in this publication.

The program stop is recirculated for one word time in the FW phase. At P_0 of FW, the K_1 and K_3 flip-flops are set, and the 0_1 flip-flop is reset. The computer therefore, goes to AW.

Logic for K_1 K_3 and $\overline{0}_1$ settings:

$$k_1 = P_0\ \overline{K}_2\ 0_1\ 0_3\ S_5$$
$$k_3 = P_0\ \overline{K}_1\ \overline{K}_2\ 0_1\ 0_3\ S_5$$
$$\overline{o}_1 = P_0\ \overline{K}_1\ \overline{K}_2\ 0_2\ 0_3\ S_5$$

The stop code of the modified store order causes the computer to remain in the AW phase for one word time while the K_2 flip-flop is set at P_0 and the logic for the stop phase is completed.

Logic for K_2 set:

$$k_2 = P_0\ K_1\ \overline{0}_1\ 0_3\ \overline{S}_4\ \overline{S}_3\ \overline{S}_2\ \overline{S}_1$$

4-67. MULTIPLY (Mu) $\overline{0}_1\ 0_2\ \overline{0}_3$ (See figures 4-20 and 4-21.)

The multiplication order forms the product (Pr) of the number in the accumulator (A_0) and the number in main memory (M_{oy}). The contents of A_0 represents the multiplier (Mr), whereas the multiplicand (Md) from M_{oy} is located by the β address of the instruction. The order originates in the Wβ phase, continues through FW and AW and is completed during LW. The number of additional word operations performed is determined by the sector address of the next instruction. This allows the multiplication operation to be terminated when the desired accuracy has been obtained in the product. The most significant digits of the product are generated first. That is, the most significant digit of the multiplier is used in the first step. In successive steps the multiplier is shifted left so that the digits are used in decreasing order of significance. The multiplicand is shifted right one bit per step, and

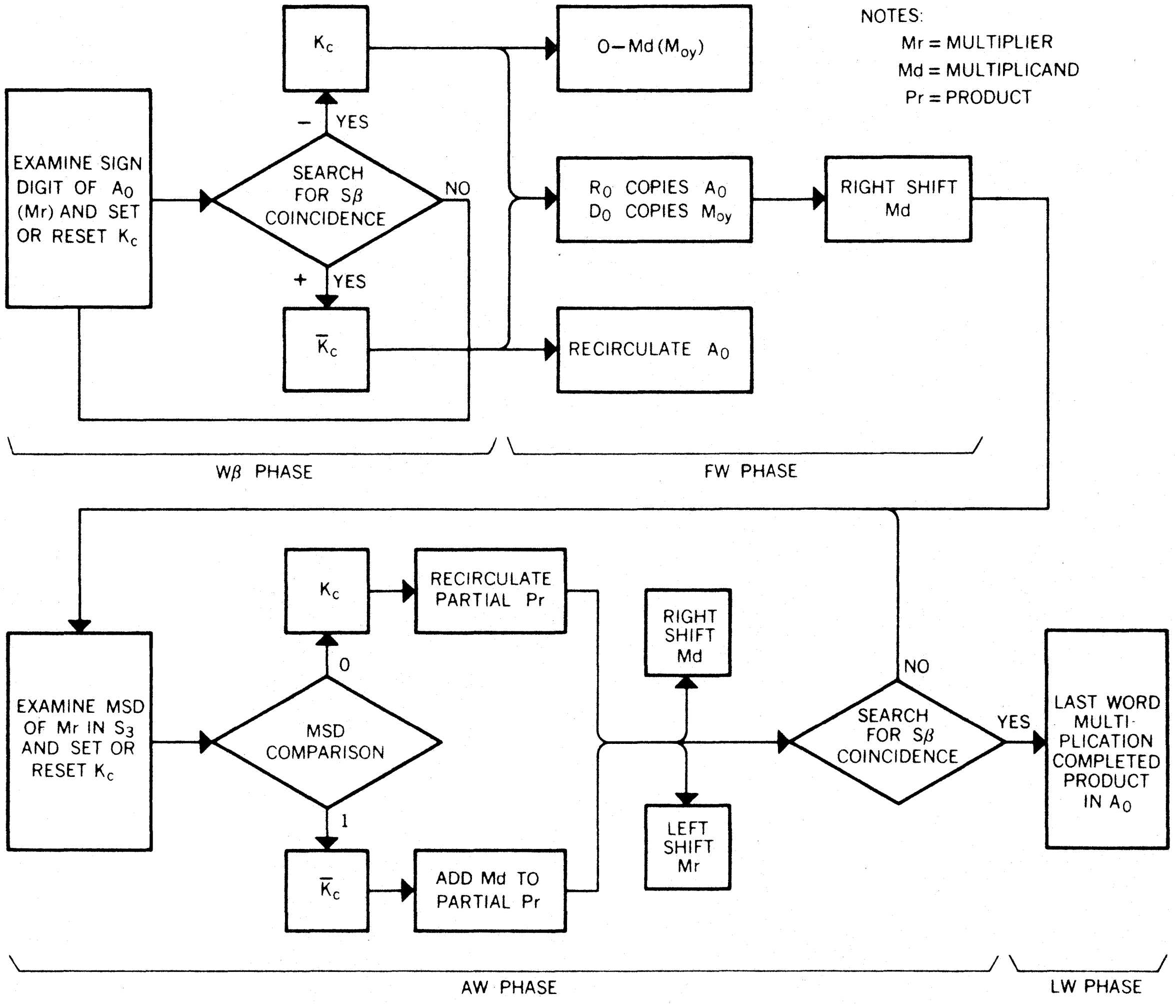


Figure 4-20. Multiply Order, Flow Diagram

a bit is dropped from the least significant end of the word for each step.

Binary multiplication is similar to decimal multiplication in the sense that:

Mr	Md	Pr = Mr Md
1	0	0
0	0	0
1	1	1
0	1	0

However, in binary multiplication the most significant digit (M.S.D.) of the multiplier is the first term used in obtaining the product.

Example:

Step 1.

$$\begin{array}{lll} \text{Md} = & 0.1\;1\;0\;1 & = 13/16 \\ \text{Mr} = & \underline{0.1^{*}0\;1\;1\quad} & = 11/16 \\ \text{Partial Pr} = & 0.0\;1\;1\;0\;1 & \end{array}$$

Arithmetic Solution
$13/16 \times 11/16 = 143/256$

* M.S.D.

After each multiplication, the Md is right shifted one bit position, and the product of the preceding multiplication is added to the partial product.

Example:

Step 2.

$$\begin{array}{ll} \text{Rt. shift Md} = & 0.0\;1\;1\;0\;1 \\ \text{Mr} = & \underline{0.1\;0\;1\;1\quad} \\ \text{Old partial Pr} = & 0.0\;1\;1\;0\;1 \\ & \underline{0.0\;0\;0\;0\;0\;0} \\ \text{New partial Pr} = & 0.0\;1\;1\;0\;1\;0 \end{array}$$

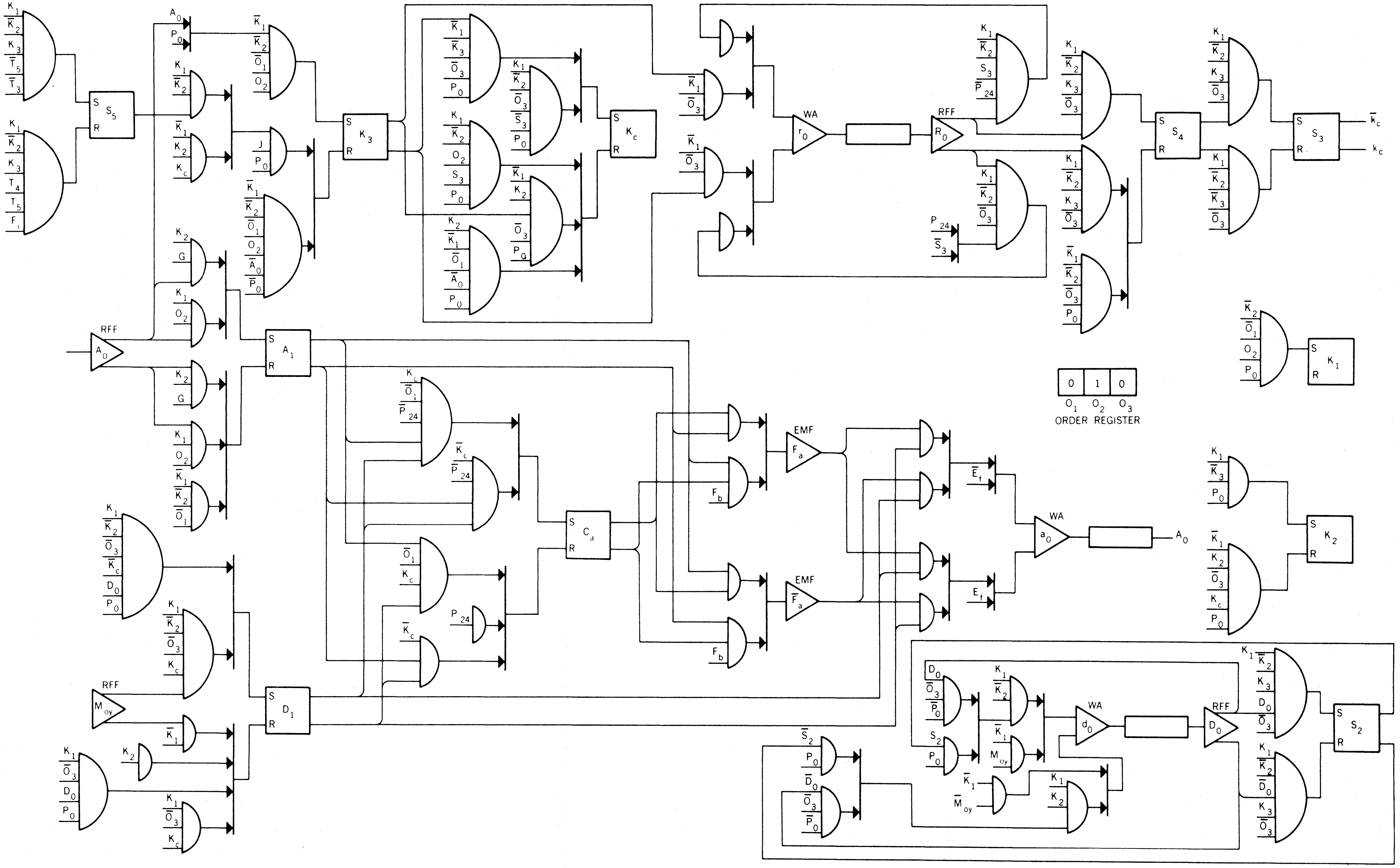


Figure 4-21. Multiply Order, Logic Diagram

Step 3.

Rt. shift Md =	0.0 0 1 1 0 1	
Mr =	0.1 0 1 1	
Old partial Pr =	0.0 1 1 0 1 0	
	0.0 0 0 1 1 0 1	
New partial Pr =	0.1 0 0 0 0 0 1	

Step 4.

Rt shift Md =	0.0 0 0 1 1 0 1	
Mr =	0.1 0 1 1	
Old partial Pr =	0.1 0 0 0 0 0 1	
	0.0 0 0 0 1 1 0 1	
Final Pr =	0.1 0 0 0 1 1 1 1	= 143/256

If, however, the multiplier is a negative number, i.e., the sign digit (digit immediately preceding the decimal) is a one, a modified type of multiplication is employed. The symbol Mr' represents the digits following the decimal point in the multiplier, and the product Mr' Md is formed. Since a one in the sign digit is equal to the quantity minus one, then the equation Mr' = Mr - (-1) = Mr + 1 is valid. Therefore, the product is equal to Mr' times Md. Which, when substituting Mr + 1 in the place of Mr', is equal to Md (Mr-1), or Mr Md + Md. Hence, to obtain a correct solution, Md must be subtracted from the product.

Example:

Md =	1.0 0 1 1	= - 13/16
Mr =	1.0 1 0 1	= - 11/16
	0.1 1 0 1 (0-Md)	
	0.0 0 0 0 0	
	1.1 1 0 0 1 1	
	0.0 0 0 0 0 0 0	
	1.1 1 1 1 0 0 1 1	
Pr =	1 0 | 0.1 0 0 0 1 1 1 1	= - 143/256
	↑ Over flow	

The multiplication order begins in the Wβ phase where the sign digit of Mr is examined at P_0 to determine whether Md is to be subtracted from 0 or not. The carry control flip-flop (K_c) is used to make this decision. If $K_c = 1$, Mr is subtracted from 0. If $K_c = 0$, A_0 is recirculated. The output of the K_c flip-flop is determined by the following input logic:

$$k_c = \bar{K}_1 K_2 \bar{T}_5 \bar{T}_3$$
$$\bar{k}_c = \bar{K}_1 K_2 K_3 F_i \left[T_5 \bar{T}_4 (\bar{T}_3 + \bar{T}_2) \right] + \bar{K}_1 K_2 \bar{0}_1 \bar{A}_0 P_0$$

The K_c flip-flop is set for Sβ coincidence and reset if no Sβ coincidence occurs. If Sβ coincidence is found, the sign bit in A_0 is copied at P_0. Where K_c is high at Sβ coincidence it remains high into the FW phase providing $A_0 P_0 = 1$. The FW phase is entered at P_0 of Wβ by resetting the K2 flip-flop.

Logic

$$\bar{k}_2 = J \bar{K}_1 K_2 (0_1 + \bar{0}_2 + \bar{0}_3) K_c P_0$$

The K_3 flip-flop is also reset at P_0 of Wβ if K_c is high at sector coincidence.

In the FW phase if K_c is one, Md is subtracted from 0. That is, $A_1 = 0$ and Md enters the adder from D_1 and the resulting operation $A_1 - D_1$, or 0 - Md, takes place. This operation is illustrated by the examples in the paragraph following step 4.

Logic for $A_1 - D_1$ (0 - Md):

$$c_a = K_c \bar{A}_1 D_1 \bar{0}_1 \bar{P}_{24}$$
$$\bar{c}_a = K_c A_1 \bar{D}_1 \bar{0}_1 + P_{24}$$
$$\bar{a}_1 = \bar{K}_1 \bar{K}_2 \bar{0}_1$$
$$d_1 = \bar{K}_1 \bar{K}_2 \bar{0}_3 K_c M_{oy}$$
$$\bar{d}_1 = \bar{K}_1 \overline{M_{oy}} + K_2$$

When M_{oy} is copied into the 24 bit register (D_0) at P_0 of FW, the multiplicand is right shifted and the least significant digit (P_{24}) is lost.

Logic for D_0:

$$\begin{aligned} d_0 &= \overline{K}_1\ M_{oy} \\ \overline{d}_0 &= \overline{K}_1\ \overline{M}_{oy} \end{aligned}$$

The K_3 flip-flop copies A_0 and contains the most significant digit at P_0 time. The K_c flip-flop copies K_3 for multiplication during the first additional word. K_3 is set at P_0 for AW, and K_c copies K_3 for FW Mu.

Logic for K_3 and K_c:

$$\begin{aligned} k_3 &= \overline{K}_1\ \overline{K}_2\ \overline{0}_1\ 0_2\ (A_0 + P_0) \\ \overline{k}_3 &= \overline{K}_1\ \overline{K}_2\ \overline{0}_1\ 0_2\ \overline{A}_0\ \overline{P}_0 \\ k_c &= \overline{K}_1\ \overline{0}_3\ \overline{K}_3\ P_0 \\ \overline{k}_c &= \overline{K}_1\ \overline{K}_2\ \overline{0}_3\ K_3\ P_0 \end{aligned}$$

In the additional word phase, K_c determines if Md or zeros are added to the partial product. The K_c flip-flop is set by S_3 which contains the most significant multiplier digit. If S_3 and K_c are reset A_1 and D_1 are added. The a_1 input signal copies A_0 for AW Mu while the $-D_1$ flip-flop copies D_0 for AW Mu if $K_c = 0$. The d_1 input copies 0 if $K_c = 1$. Hence, if $K_c = 0$, A_1 and D_1 are added, and C_a is reset at P_{24}. When S_3 is 0 and K_c is high, no operation takes place.

Logic for C_a, K_c, A_1 and D_1 flip-flops:

$$\begin{aligned} c_a &= \overline{K}_c\ A_1\ D_1\ \overline{P}_{24} \\ \overline{c}_a &= \overline{K}_c\ \overline{A}_1\ \overline{D}_1 + P_{24} \end{aligned}$$

$$\begin{aligned} k_c &= K_1\ \overline{K}_2\ \overline{0}_3\ \overline{S}_3\ P_0 \\ \overline{k}_c &= K_1\ \overline{K}_2\ 0_2\ S_3\ P_0 \end{aligned}$$

$$\begin{aligned} a_1 &= K_1\ 0_2\ A_0 \\ \overline{a}_1 &= K_1\ 0_2\ \overline{A}_0 \end{aligned}$$

$$\begin{aligned} d_1 &= K_1\ \overline{K}_2\ \overline{0}_3\ \overline{K}_c\ D_0\ \overline{P}_0 \\ \overline{d}_1 &= K_1\ (\ \overline{0}_3\ \overline{D}_0\ \overline{P}_0 + \overline{0}_3\ K_c) \end{aligned}$$

The multiplicand is right shifted in D_0 by means of a 24 bit register. The sign digit is extended from S_2 on P_0 of each word. Since the L.S.D., or P_{24}, was lost during the right shift in FW, the sign digit is now in R_0 at P_1 time in the first additional word. However, S_2 copies D_0 so that the sign digit is contained at P_0. The d_0 input to the D_0 flip-flop copies S_2 for the extended sign digit at P_0. For all the other 23 bits, d_0 copies the D_0 recirculation register. When d_0 copies S_2, another digit on the least significant end is lost. At the end of 24 additional words the D_0 register contains all ones or all zeros, depending upon the sign digit.

Logic for S_2 and D_0:

$$s_2 = K_1 \overline{K}_2 K_3 \overline{0}_3 D_0$$
$$\overline{s}_2 = K_1 \overline{K}_2 K_3 \overline{0}_3 \overline{D}_0$$

$$d_0 = K_1 \overline{K}_2 (S_2 P_0 + \overline{0}_3 D_0 \overline{P}_0)$$
$$\overline{d}_0 = K_1 \overline{K}_2 (\overline{S}_2 P_0 + \overline{0}_3 \overline{D}_0 \overline{P}_0)$$

The multiplier is left shifted in R_0 so that the most significant digit may be used as the multiplier. This is accomplished by using a 26 bit register which is formed by the R_0 register and the S_3 and S_4 flip-flops. The sign digit of Mr was lost at P_0 of FW when K_3 was set for AW. At P_{24} of the first AW, the 26 bit register contains the following: P_1 - P_{24} in R_0, P_0 in S_4, and P_1 in S_3. The S_4 flip-flop was reset at P_0 of FW and a zero was shifted into the register at P_{24} instead of recirculating the used multiplier digit contained in S_3. At P_0 of the first AW, S_3 contains the next significant Mr digit (P_2). All succeeding Mr bits are contained in S_3 at P_0 of each additional word. At the end of 24 additional words the R_0 register contains all zeros.

Logic for the 26 bit register:

$$s_4 = K_1 \overline{K}_2 K_3 \overline{0}_3 R_0$$
$$\overline{s}_4 = K_1 \overline{K}_2 K_3 \overline{0}_3 \overline{R}_0 + \overline{K}_1 \overline{K}_2 P_0 \overline{0}_3$$

$$s_3 = K_1 \overline{K}_2 K_3 \overline{0}_3 S_4$$
$$\overline{s}_3 = K_1 \overline{K}_2 K_3 \overline{0}_3 \overline{S}_4$$

$$r_0 = K_1 \overline{K}_2 S_3 \overline{P}_{24}$$
$$\overline{r}_0 = K_1 \overline{K}_2 \overline{0}_3 (\overline{S}_3 + P_{24})$$

For the R_0 register S_4 copies R_0 for AW Mu and is reset at P_0 of FW. The S_3 flip-flop copies S_4 during AW Mu, while R_0 copies S_3 except at P_{24}. At P_{24} time $r_0 = 0$.

To go from AW to LW beta sector coincidence is used. Since the R_0 contains all zeros after 24 additional words no further multiplication takes place. Therefore, sector coincidence may be anywhere on the memory drum. In this instance S_5 is used as the sector coincidence detector instead of K_c. The S_5 flip-flop is set at P_{21} or P_{16} of AW. The S_5 flip-flop is reset if no coincidence is detected during AW Sα (P_8 - P_3). If sector coincidence occurs and S_5 is set, K_3 is reset at P_0 and the LW phase is entered.

Logic for S_5 and $\overline{K}_3$:

$$s_5 = K_1 \overline{K}_2 K_3 \overline{T}_5 \overline{T}_3$$
$$\overline{s}_5 = K_1 \overline{K}_2 K_3 T_5 T_4 F_i$$
$$\overline{k}_3 = K_1 \overline{K}_2 S_5 P_0 J$$

In the last word phase all operations continue exactly as in AW with one term of multiplication being completed at the end of LW. The product is located in A_0, while D_0 contains either zeros or ones as stipulated by the sign of Md, and R_0 contains all zeros. The K_2 flip-flop is set at P_0 of LW thereby returning the computer to the IR phase.

Logic for K_2:

$$k_2 = P_0 K_1 \overline{K}_3$$

4-68. DIVIDE ORDER (Dv) $0_1\ \overline{0}_2\ 0_3$ (See figures 4-22 and 4-23.)

The divide order forms the quotient of the dividend and the divisor. The dividend is stored in the A_0 register from a previous operation. The divisor is selected from main memory via M_{oy} and is read into the D_0 register during the FW phase. The quotient is developed in the R_0 register, one quotient bit per word time starting with the sign bit. The length of the division order, or the number of quotient bits generated, is determined by the program through sector coincidence termination of the AW phase. During the LW phase the quotient is rounded-off. Whether the round-off entails an addition or subtraction is determined during the last word time of the AW phase. The round-off occurs as the quotient is transferred from the R_0 register to the A_0 register.

Initially, during the FW phase', the R_0 register is cleared to all zeros, then an index bit is inserted at P_0 of FW. Also during P_0 of FW, the sign bits of the dividend and the divisor are compared and the result, as shown in figure 4-25, is stored in the K_c flip-flop for use during the following word time. During the first word time of AW, the bit stored in K_c (sign bit of the quotient) is inserted in the R_0 register one bit time after the reappearance of the index bit. The index bit reappears at P_{24} time; therefore, the quotient bit is stored at P_0 time. The index bit is then right shifted one bit position so that the next quotient bit generated will be stored at P_1 time. The quotient bits are not shifted, but simply recirculated. This process continues throughout the AW phase. The sign of the quotient in the R_0 register is actually the reverse of the proper sign; however, during the LW phase the sign bit is again reversed to yield the proper sign.

Meanwhile, during AW, the dividend (or initial remainder) is left shifted one bit position each word time. The divisor is simply recirculated in the D_0 register and added to or subtracted from the remainder as determined by the state of the K_c flip-flop. At the completion of each addition or subtraction, the result of the comparison of the most significant bits (sign bits) of the remainder and the divisor is again stored in the K_c flip-flop to control the adder circuits during the next word time, and to

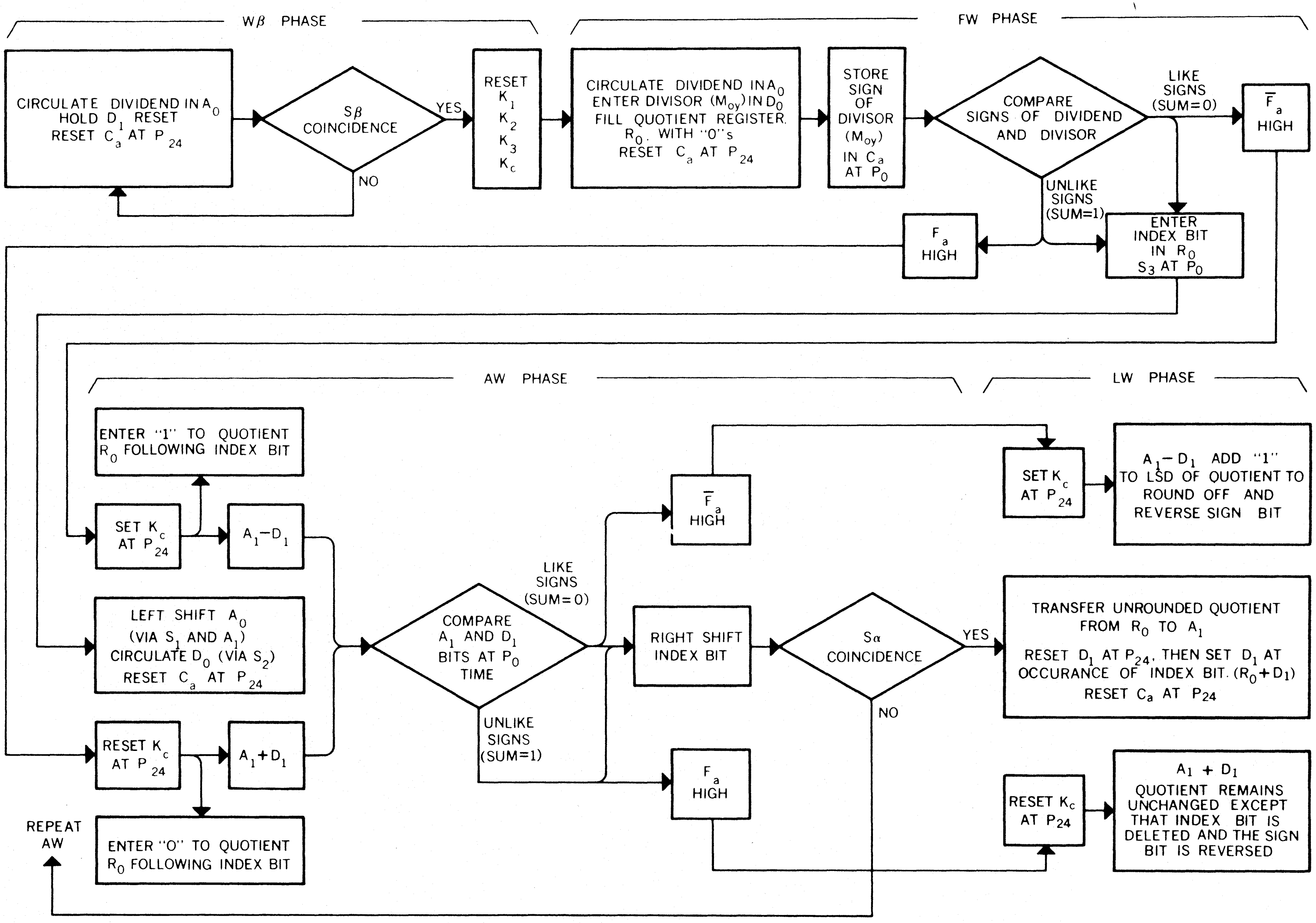
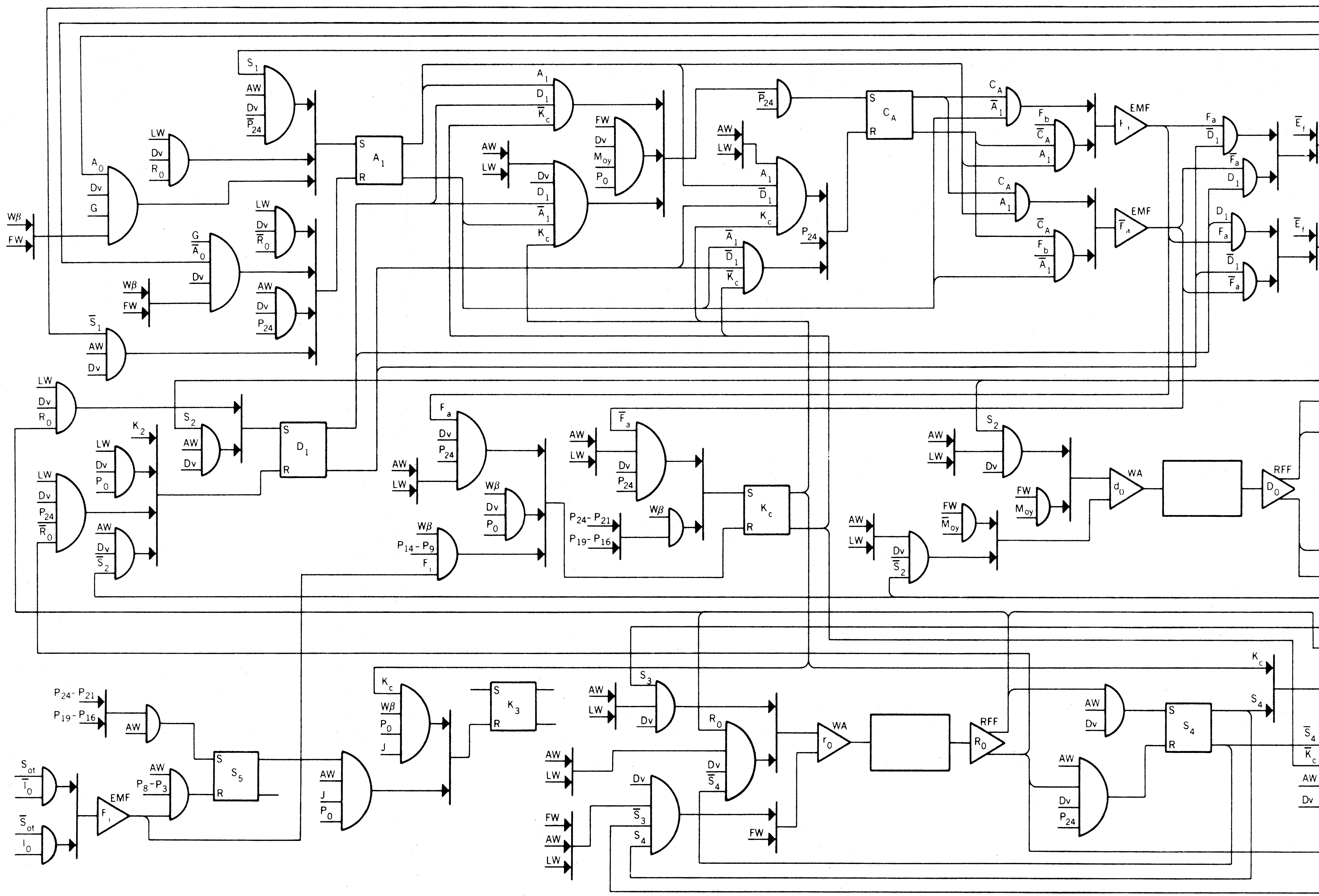
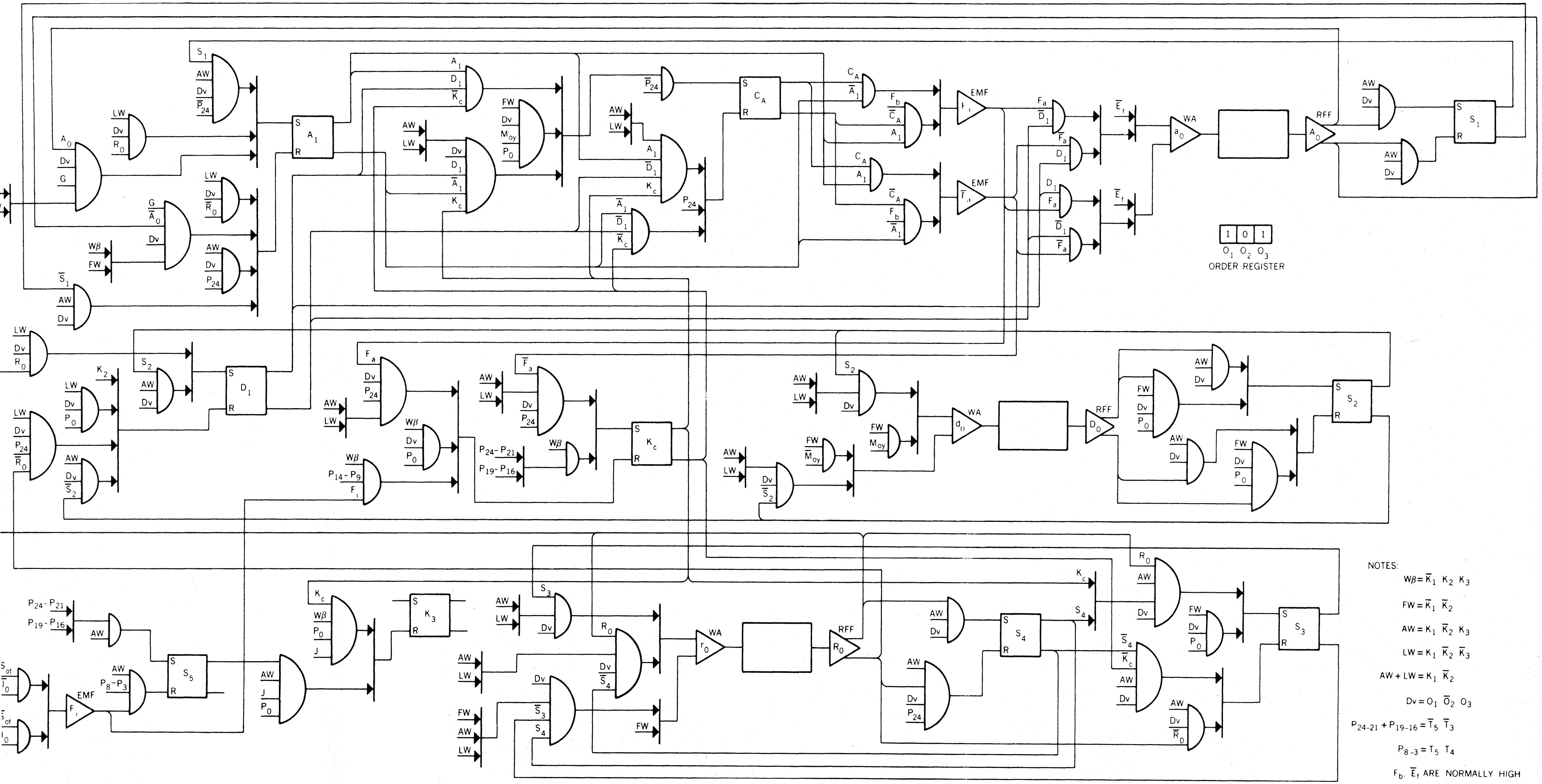

Wβ PHASE
CIRCULATE DIVIDEND IN A_0 HOLD D_1 RESET RESET C_a AT P_{24}
Sβ COINCIDENCE
YES
NO
RESET K_1 K_2 K_3 K_c
FW PHASE
CIRCULATE DIVIDEND IN A_0 ENTER DIVISOR (M_{oy}) IN D_0 FILL QUOTIENT REGISTER. R_0. WITH "0"s RESET C_a AT P_{24}
STORE SIGN OF DIVISOR (M_{oy}) IN C_a AT P_0
COMPARE SIGNS OF DIVIDEND AND DIVISOR
LIKE SIGNS (SUM=0)
$\overline{F}_a$ HIGH
UNLIKE SIGNS (SUM=1)
F_a HIGH
ENTER INDEX BIT IN R_0 S_3 AT P_0
AW PHASE
LW PHASE
ENTER "1" TO QUOTIENT R_0 FOLLOWING INDEX BIT
SET K_c AT P_{24}
$A_1 - D_1$
LEFT SHIFT A_0 (VIA S_1 AND A_1) CIRCULATE D_0 (VIA S_2) RESET C_a AT P_{24}
RESET K_c AT P_{24}
$A_1 + D_1$
REPEAT AW
ENTER "0" TO QUOTIENT R_0 FOLLOWING INDEX BIT
COMPARE A_1 AND D_1 BITS AT P_0 TIME
LIKE SIGNS (SUM=0)
UNLIKE SIGNS (SUM=1)
$\overline{F}_a$ HIGH
RIGHT SHIFT INDEX BIT
F_a HIGH
Sα COINCIDENCE
YES
NO
SET K_c AT P_{24}
$A_1 - D_1$ ADD "1" TO LSD OF QUOTIENT TO ROUND OFF AND REVERSE SIGN BIT
TRANSFER UNROUNDED QUOTIENT FROM R_0 TO A_1
RESET D_1 AT P_{24}, THEN SET D_1 AT OCCURANCE OF INDEX BIT. (R_0+D_1)
RESET C_a AT P_{24}
RESET K_c AT P_{24}
$A_1 + D_1$ QUOTIENT REMAINS UNCHANGED EXCEPT THAT INDEX BIT IS DELETED AND THE SIGN BIT IS REVERSED

Figure 4-22. Divide Order, Flow Diagram



S1
AW
Dv
$\overline{P}_{24}$
LW
Dv
R_0
A_0
Dv
G
Wβ
FW
G
$\overline{A}_0$
Dv
LW
Dv
$\overline{R}_0$
AW
Dv
P_{24}
Wβ
FW
$\overline{S}_1$
AW
Dv
S
A_1
R
A_1
D_1
$\overline{K}_c$
AW
LW
Dv
D_1
$\overline{A}_1$
K_c
FW
Dv
M_{oy}
P_0
$\overline{P}_{24}$
AW
LW
A_1
$\overline{D}_1$
K_c
P_{24}
$\overline{A}_1$
$\overline{D}_1$
$\overline{K}_c$
S
C_A
R
C_A
$\overline{A}_1$
F_b
$\overline{C}_A$
A_1
C_A
A_1
$\overline{C}_A$
F_b
$\overline{A}_1$
EMF
F_f
EMF
$\overline{F}_f$
F_a
$\overline{D}_1$
$\overline{F}_a$
D_1
D_1
F_a
$\overline{D}_1$
$\overline{F}_a$
$\overline{E}_f$
$\overline{E}_f$
LW
Dv
R_0
K_2
S_2
AW
Dv
S
D_1
R
LW
Dv
P_0
LW
Dv
P_{24}
$\overline{R}_0$
AW
Dv
$\overline{S}_2$
F_a
Dv
P_{24}
AW
LW
Wβ
Dv
P_0
Wβ
$P_{14}-P_9$
F_f
$\overline{F}_a$
AW
LW
Dv
P_{24}
$P_{24}-P_{21}$
$P_{19}-P_{16}$
Wβ
S
K_c
R
S_2
AW
LW
Dv
FW
M_{oy}
FW
$\overline{M}_{oy}$
AW
LW
Dv
$\overline{S}_2$
WA
d_0
RFF
D_0
$P_{24}-P_{21}$
$P_{19}-P_{16}$
AW
S_{ot}
$\overline{I}_0$
$\overline{S}_{ot}$
I_0
EMF
F_f
AW
P_8-P_3
S
S_5
R
K_c
Wβ
P_0
J
AW
J
P_0
S
K_3
R
S_3
AW
LW
Dv
R_0
AW
LW
Dv
$\overline{S}_4$
Dv
$\overline{S}_3$
S_4
FW
AW
LW
FW
WA
r_0
RFF
R_0
AW
Dv
S
S_4
R
AW
Dv
P_{24}
K_c
S_4
$\overline{S}_4$
$\overline{K}_c$
AW
Dv

Figure 4-2



NOTES:

$W\beta = \overline{K}_1 \; K_2 \; K_3$

$FW = \overline{K}_1 \; \overline{K}_2$

$AW = K_1 \; \overline{K}_2 \; K_3$

$LW = K_1 \; \overline{K}_2 \; \overline{K}_3$

$AW + LW = K_1 \; \overline{K}_2$

$Dv = O_1 \; \overline{O}_2 \; O_3$

$P_{24-21} + P_{19-16} = \overline{T}_5 \; \overline{T}_3$

$P_{8-3} = T_5 \; T_4$

F_b, $\overline{E}_f$ ARE NORMALLY HIGH

Figure 4-23. Divide Order, Logic Diagram

enter the new quotient bit in the D_0 register. This process continues until sector coincidence occurs as determined by the program and the computer enters the LW phase.

During the one word time of the LW phase, the generated quotient in the R_0 register is shifted into the A_0 register. As it is being shifted, it passes through the adder circuit where D_1 is either added or subtracted. Flip-flop D_1 is initially (at P_{24} time) reset, and then is set by the occurrence of the index bit from the R_0 register and remains set until reset by P_0. If K_c is in the set state (again, as the result of previous comparison of the divisor and the remainder) D_1 is subtracted from the quotient A_1 which effectively adds a one to the last generated quotient bit. If K_c is in the reset state, D_1 is added to the quotient (A_1) which effectively leaves the quotient unchanged except for deleting the index bit. In either event (K_c or $\overline{K}_c$) the sign of the quotient is reversed to provide the correct sign bit. The rounded quotient recirculates in the A_0 register.

The following example illustrates the division process:

Given: dividend = 1.101000 = -3/8
divisor = 1.100000 = -1/2

$$\begin{array}{r} 1.1011 \\ 1.100000\ \overline{)\,1.101000} \end{array}$$

		Compare sign bits: Like ∴ set K_c.
	1.010000	Left shift dividend (initial).
(-)	1.100000	K_c is set ∴ quotient bit = 1. ∴ subtract divisor.
	1.110000	Compare sign bits: Like ∴ set K_c.
	1.100000	Left shift dividend (remainder).
(-)	1.100000	K_c is set ∴ Q = 1; subtract divisor.
	0.000000	Compare sign bits: Unlike ∴ reset K_c.
	0.000000	Left shift dividend (remainder).
(+)	1.100000	K_c is reset ∴ Q = 0; add divisor.
	1.100000	Compare sign bits: Like ∴ set K_c.
	1.000000	Left shift dividend (remainder).

(-) $\underline{1.100000}$ K_c is set ∴ Q = 1; subtract divisor.

1.100000 Compare sign bits: Like ∴ set K_c.

1.000000 Left shift dividend (remainder).

(-) $\underline{1.100000}$ K_c is set ∴ Q = 1; subtract divisor.

1.100000 Compare sign bits: Like ∴ set K_c.

Sector coincidence occurs

Unrounded quotient = 1.10111* (A_1)

Round-off quantity = $\underline{1.00001}$ (D_1) K_c is set. Subtract D_1.

Rounded quotient 0.11000 = +3/4

* In the unrounded quotient the index bit appears immediately after the last quotient digit. During the AW phase, the index bit is shifted to the right one bit position each word time.

The following paragraphs provide a detailed logic description of the divide order:

During the Wβ phase the K_C, D_1, C_a, and K_3 flip-flops are assigned their proper states in preparation for the division order. The K_C flip-flop is set at P_{24} and remains high when Sβ coincidence is reached. At P_0 the K_c flip-flop is reset. The D_1 flip-flop, which was reset by K_2 during the Wα , IR, and Wβ phases, remains low during the FW phase since no d_1 logic occurs to set the flip-flop. As a result, the carry flip-flop C_a which was reset at P_{24}, remains low as long as D_1 is reset. When sector beta coincidence occurs K_3 is reset.

Logic:

$$\overline{k}_c = \overline{K}_1 K_2 P_0 (\overline{0}_2 + 0_3)$$
$$\overline{d}_1 = K_2$$
$$\overline{c}_a = P_{24}$$
$$\overline{k}_3 = \overline{K}_1 K_2 K_C$$

In the FW phase, the dividend (initial remainder) is recirculated in A_0 and A_1 by the following logic:

$$\begin{aligned}
a_1 &= G\,\overline{K}_1\,0_1\,A_0\,F_j \\
\overline{a}_1 &= G\,\overline{K}_1\,0_1\,\overline{A}_0\,F_j \\
a_0 &= \overline{E}_f\,\overline{D}_1\,\overline{C}_a\,A_1\,F_b \\
\overline{a}_0 &= \overline{E}_f\,\overline{D}_1\,\overline{C}_a\,\overline{A}_1\,F_b
\end{aligned}$$

Note that $\overline{D}_1$ and $\overline{C}_a$ are high.

The divisor is read into D_0 from M_{oy}, and the least significant digit is read into S_2 from D_0 at P_0 while the sign digit is read into C_a.

Logic:

$$\begin{aligned}
d_0 &= \overline{K}_1\,M_{oy} \\
\overline{d}_0 &= \overline{K}_1\,\overline{M}_{oy} \\
s_2 &= \overline{K}_1\,\overline{K}_2\,\overline{0}_2\,D_0\,P_0 \\
\overline{s}_2 &= \overline{K}_1\,\overline{K}_2\,\overline{0}_2\,\overline{D}_0\,P_0 \\
c_a &= \overline{K}_1\,\overline{K}_2\,0_1\,\overline{0}_2\,0_3\,\overline{P}_{24}\,M_{oy}\,P_0 \\
\overline{c}_a &= P_{24}
\end{aligned}$$

R_0 is cleared to all zeros and S_3 is set (index bit).

Logic:

$$\begin{aligned}
\overline{r}_0 &= \overline{K}_1\,\overline{K}_2\,\overline{K}_3 \\
s_3 &= \overline{K}_1\,\overline{K}_2\,\overline{0}_2\,P_0
\end{aligned}$$

During the first AW phase the index bit is written in r_0 at P_{24} by S_3. The S_3 flip-flop was set at P_0 of FW and reset when the first zero occurs on the R_0 register.

Logic:

$$\begin{aligned}
r_0 &= K_1\,\overline{K}_2\,S_3\,0_3 \\
s_3 &= K_1\,\overline{K}_2\,K_3\,\overline{R}_0\,\overline{0}_2
\end{aligned}$$

The index digit, written into Y_0 at P_{24} of the first additional word, is read out of R_0 24 bits later at P_0, and rewritten into r_0. This action right shifts the index digit one bit for each additional word. The $\overline{S}_4$ gate allows only the index digit to shift.

Logic:

$$\begin{aligned}
r_0 &= K_1\, \overline{K}_2\, 0_3\, \overline{S}_4\, R_0 \\
\overline{s}_4 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, \overline{R}_0\, P_{24} \\
s_4 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, R_0
\end{aligned}$$

Note S_4 is set by R_0 (index digit).

The quotient digit is read into K_c at P_{24} and stored until the next quotient digit. The S_3 flip-flop copies K_c at R_0 and reads into r_0 one bit later. The S_3 flip-flop then reads R_0 to recirculate all previous quotient digits placed in R_0.

Logic:

$$\begin{aligned}
s_3 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, R_0\, (K_c + S_4) \\
\overline{s}_3 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, (\overline{S}_4\, \overline{K}_c + \overline{R}_0) \\
r_0 &= K_1\, \overline{K}_2\, 0_3\, S_3 \\
\overline{r}_0 &= \overline{K}_2\, 0_3\, \overline{S}_3\, (S_4 + \overline{R}_0) \\
s_4 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, R_0 \\
\overline{s}_4 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, \overline{R}_0\, P_{24}
\end{aligned}$$

The quotient digit is generated by comparing D_1 and a_0 at P_{24}. The divisor in D_0 is recirculated through S_2 to form a 25 bit register which is copied by D_1. The D_1 flip-flop provides the sign digit of the divisor to the serial adder at P_{24} due to the one bit delay through S_2.

Logic:

$$\begin{aligned}
s_2 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, D_0 \\
\overline{s}_2 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, \overline{D}_0 \\
d_0 &= K_1\, \overline{K}_2\, 0_3\, S_2 \\
\overline{d}_0 &= K_1\, \overline{K}_2\, 0_3\, \overline{S}_2 \\
d_1 &= K_1\, \overline{K}_2\, K_3\, \overline{0}_2\, S_2 \\
\overline{d}_1 &= K_1\, K_3\, \overline{0}_2\, \overline{S}_2
\end{aligned}$$

For each additional word, A_0 is left shifted and a zero is shifted into the least significant digit position. The left shifted quantity, or remainder, is read into the serial adder. The left shift modification is accomplished by a two bit delay provided by S_1, which copies A_0, and by A_1, which copies S_1. This 26 bit register places each new sign digit of the remainder in the adder at P_{24}.

Logic:

$$\begin{aligned} s_1 &= K_1 \overline{K}_2 K_3 \overline{0}_2 A_0 \\ \overline{s}_1 &= K_1 \overline{K}_2 K_3 \overline{0}_2 \overline{A}_0 \\ a_1 &= K_1 \overline{K}_2 K_3 \overline{0}_2 S_1 \overline{P}_{24} \\ \overline{a}_1 &= K_1 \overline{K}_2 K_3 \overline{0}_2 (\overline{S}_1 + P_{24}) \end{aligned}$$

Note: The a_0 logic along with the C_a logic is the normal add - subtract logic, with the K_c carry control flip-flop.

The quotient digits generated by comparing D_1 and a_0 at P_{24} are stored in K_c via f_a and $\overline{f}_a$. The state of the K_c flip-flop hinges on the result of the comparison between the divisor and the remainder. That is, when the sign digits of the divisor and the remainder are equal, K_c is set and the action of the serial adder is that of normal subtract $(A_1 - D_1)$. However, when the sign digits of the divisor and the remainder are not the same, K_c is reset and the addition function of the serial adder takes place.

Logic:

$$\begin{aligned} k_c &= K_1 \overline{K}_2 \overline{0}_2 \overline{F}_a P_{24} \\ \overline{k}_c &= K_1 \overline{K}_2 0_3 F_a P_{24} \end{aligned}$$

The K_c logic is derived from the following:

$$\begin{aligned} k_c &= (a_0 D_1 + \overline{a}_0 \overline{D}_1) P_{24} \\ a_0 &= F_a \overline{D}_1 + \overline{F}_a D_1 \\ \overline{a}_0 &= \overline{F}_a \overline{D}_1 + \overline{F}_a \overline{D}_1 \end{aligned}$$

Substituting:

$$\begin{aligned} k_c &= (F_a \overline{D}_1 + \overline{F}_a D_1) D_1 P_{24} + (F_a D_1 + \overline{F}_a \overline{D}_1) \overline{D}_1 P_{24} \\ &= (F_a \overline{D}_1 D_1 + \overline{F}_a D_1 D_1 + F_a D_1 \overline{D}_1 + \overline{F}_a \overline{D}_1 \overline{D}_1) P_{24} \end{aligned}$$

Since: $D_1 \bar{D}_1 = 0$ and $D_1 D_1 = D_1$

$$k_c = (\bar{F}_a D_1 + \bar{F}_a \bar{D}_1) P_{24} = \bar{F}_a (D_1 + \bar{D}_1) P_{24}$$

and: $D_1 + \bar{D}_1 = 1$

∴ $k_c = \bar{F}_a P_{24}$

The term $K_1 \bar{K}_2 \bar{0}_2$ defines AW or LW division.

The bar side of the K_c logic is derived by the same procedure.

$$\bar{k}_c = (a_0 \bar{D}_1 + \bar{a}_0 D_1) P_{24}$$

Substituting:

$$\bar{k}_c = (F_a \bar{D}_1 + \bar{F}_a D_1) \bar{D}_1 P_{24} + (F_a D_1 + \bar{F}_a \bar{D}_1) D_1 P_{24}$$

$$= (F_a \bar{D}_1 \bar{D}_1 + \bar{F}_a D_1 \bar{D}_1 + F_a D_1 D_1 + \bar{F}_a \bar{D}_1 D_1) P_{24}$$

$$\bar{k}_c = F_a (\bar{D}_1 + D_1) P_{24}$$

∴ $\bar{k}_c = F_a P_{24}$

The term $K_1 \bar{K}_2 0_3$ defines AW or LW division.

The programming of S*a* coincidence in the additional words phase determines the length of the division. Since K_c is used to store the quotient digit and for add or subtract control, S_5 is used for coincidence detection.

Logic:

$$s_5 = K_1 \bar{K}_2 K_3 \bar{T}_5 \bar{T}_3$$

$$\bar{s}_5 = K_1 \bar{K}_2 K_3 T_5 T_4 F_i$$

$$f_i = S_{ot} \bar{I}_0 + \bar{S}_{ot} I_0$$

After S*a* coincidence, AW goes to LW at P_0 by resetting the K_3 flip-flop.

Logic:

$$\bar{k}_3 = J K_1 \bar{K}_2 S_5 P_0$$

During the LW phase, A_1 copies the unrounded quotient of R_0.

Logic:

$$a_1 = K_1 \overline{K}_2 \overline{K}_3 \overline{0}_2 R_0$$
$$\overline{a}_1 = K_1 \overline{K}_2 \overline{K}_3 \overline{0}_2 \overline{R}_0$$

The D_1 flip-flop which is reset at P_{24}, is set by the index digit and then reset again at P_0. This provides the value required to round-off the quotient in R_0.

Logic:

$$\overline{d}_1 = K_1 \overline{K}_3 \overline{0}_2 \overline{R}_0 P_{24}$$
$$d_1 = K_1 \overline{K}_2 \overline{K}_3 \overline{0}_2 R_0$$
$$\overline{d}_1 = K_1 \overline{K}_3 \overline{0}_2 P_0$$

This D_1 logic supplies the round off quantity to the serial adder. This value is then added to, or subtracted from, the unrounded quotient.

4-69. SIGMATOR

The sigmator, comprising a portion of the computer, provides an accurate, high-speed method of summing, storing, and processing incremental inputs from external sources. Typical uses of the sigmator are: accepting and summing random pulses, integrating various functions under the control of real time, transmitting data link information, and accepting the output of the analog to digital shaft converter. The incremental input flip-flops are also used as a shift register and a count-down register. All functions are performed during normal operation of the computer. Logic control divides the operation of the sigmator into eight phases which are described later in this section.

The major portions of the sigmator are the two tracks on the memory drum. One track is the pulse accumulator, or sigmator short line, and the other is the integrand register, or sigmator long line. Information is processed to and from the tracks by logic-controlled write and read heads and amplifiers together with associated flip-flops and emitter followers. Figure 4-24 shows the sequential operation of the sigmator. A functional description of each of the major portions of the sigmator is contained in the following paragraphs and is followed by a description of each phase of operation. Detailed logic for the sigmator is shown in figure 4-25.

4-70. Major Portions of the Sigmator

4-71. Pulse Accumulator Pulse Hold Flip-Flops. Incremental input pulses are stored temporarily in flip-flops V_1 through V_6 and T_s. Pulse time control logic qualifies the output of the flip-flops into the incremental adder.

4-72. Incremental Adder (One-Bit Adder-Subtractor). The flip-flops C_4 and C_5 function as a "carry" (+ register), and a "borrow" (- register) respectively, to add algebraically the input increments. The incremental adder emitter follower F_c, and the normal pulse accumulator emitter follower F_v feed into the pulse accumulator through the write amplifier and head x_0.

4-73. Pulse Accumulator (Sigmator Short Line). The pulse accumulator, or sigmator short line, consists of a track on the memory drum, an associated write head, and five read heads. The write head x_0 records the output from emitter followers F_c and X_{o1}. Read head X_{o1}, spaced one word time later than x_0, supplies the incremental adder with accumulated increments which are added to the new input increment. The output of the pulse accumulator is read from one of four read heads, spaced around the drum, time-sharing the read flip-flop X_{o2}.

4-74. Serial Adder Input Selector. The serial adder input selector consists of logic including the phase control register, which selects the output from either the pulse accumulator, via read head X_{o2}, or the integrand register, via Y_{o1} or Y_{o2} read heads. The output of the serial adder input selector is fed to the serial full adder through emitter followers F_1 and F_2.

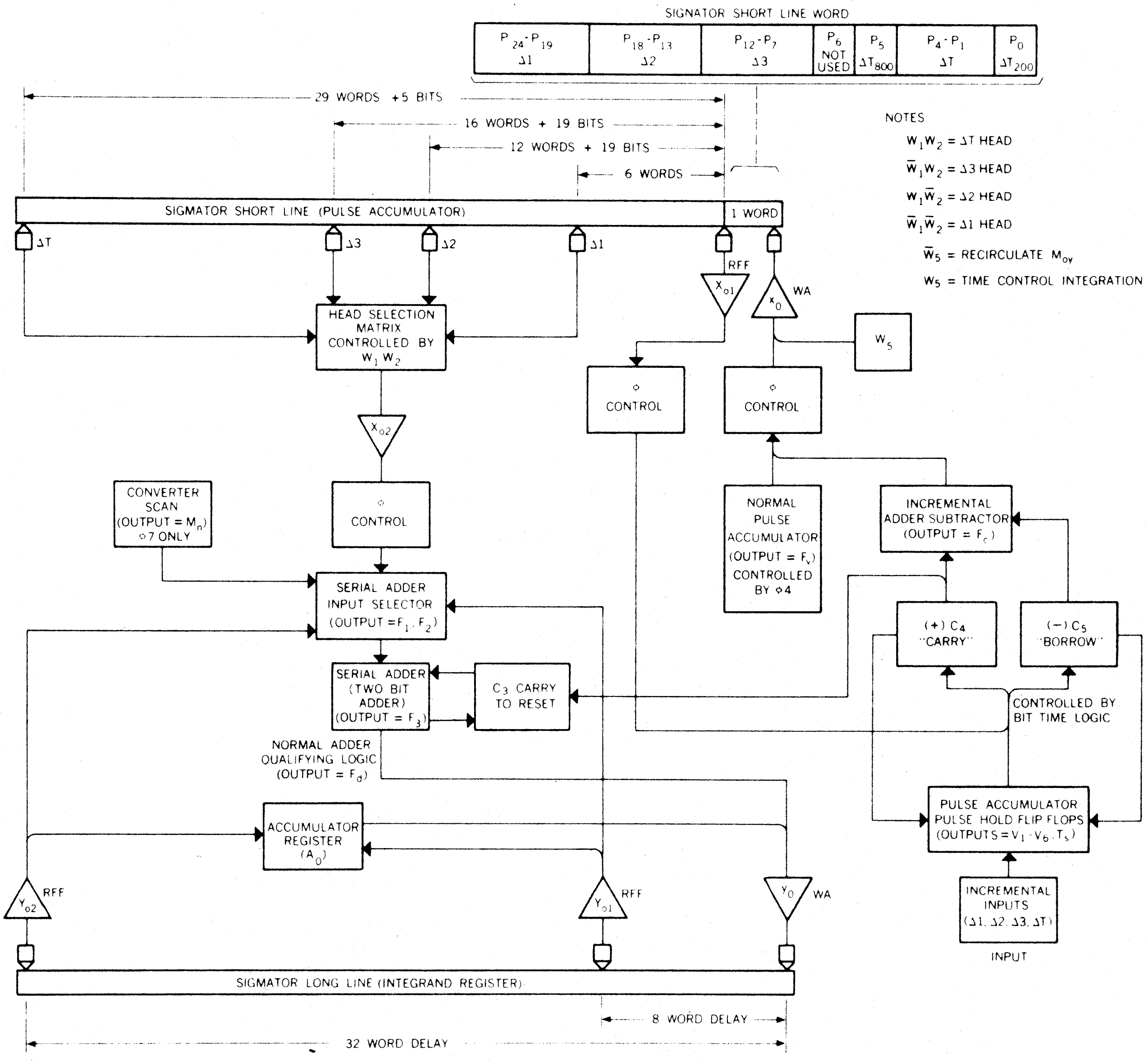
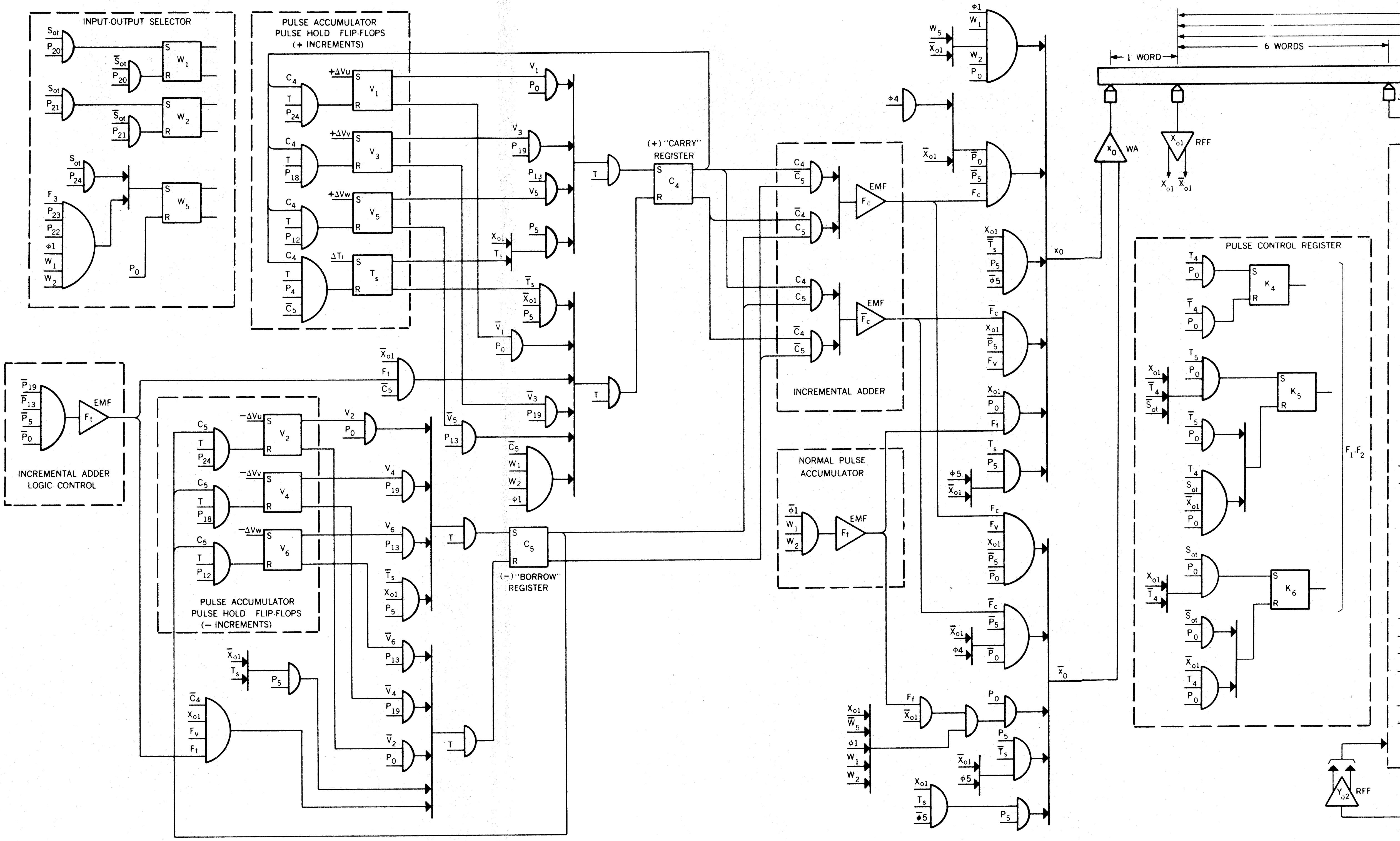
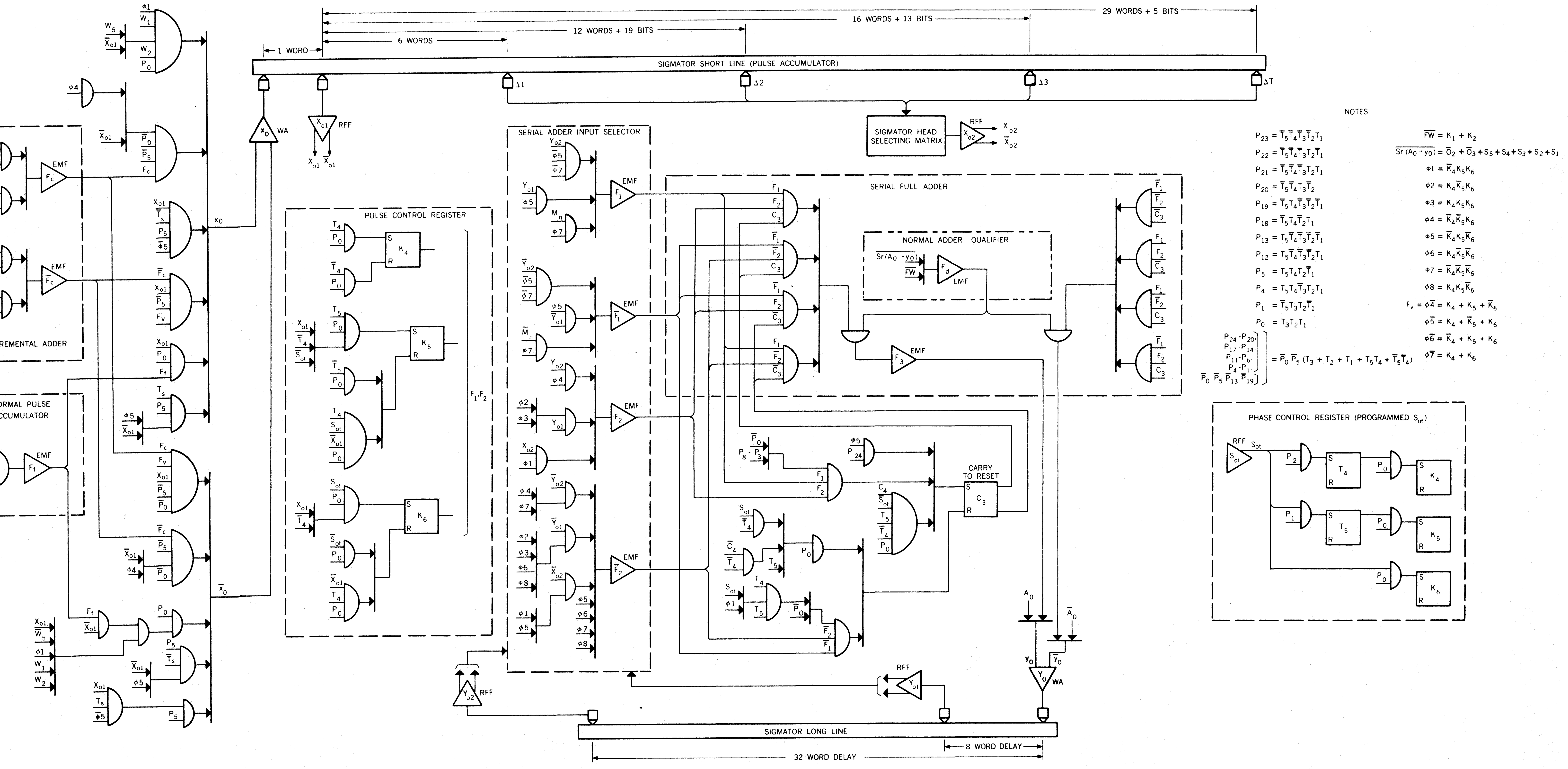

SIGNATOR SHORT LINE WORD
P_{24}-P_{19} Δ1
P_{18}-P_{13} Δ2
P_{12}-P_7 Δ3
P_6 NOT USED
P_5 $ΔT_{800}$
P_4-P_1 ΔT
P_0 $ΔT_{200}$
29 WORDS +5 BITS
16 WORDS + 19 BITS
12 WORDS + 19 BITS
6 WORDS
NOTES
W_1W_2 = ΔT HEAD
$\bar{W}_1W_2$ = Δ3 HEAD
$W_1\bar{W}_2$ = Δ2 HEAD
$\bar{W}_1\bar{W}_2$ = Δ1 HEAD
$\bar{W}_5$ = RECIRCULATE M_{oy}
W_5 = TIME CONTROL INTEGRATION
SIGMATOR SHORT LINE (PULSE ACCUMULATOR)
1 WORD
ΔT
Δ3
Δ2
Δ1
RFF
X_{o1}
x_0
WA
W_5
HEAD SELECTION MATRIX CONTROLLED BY W_1 W_2
φ CONTROL
φ CONTROL
X_{o2}
φ CONTROL
CONVERTER SCAN (OUTPUT = M_n) φ7 ONLY
NORMAL PULSE ACCUMULATOR (OUTPUT = F_v) CONTROLLED BY φ4
INCREMENTAL ADDER SUBTRACTOR (OUTPUT = F_c)
SERIAL ADDER INPUT SELECTOR (OUTPUT = F_1, F_2)
(+) C_4 "CARRY"
(−) C_5 "BORROW"
SERIAL ADDER (TWO BIT ADDER) (OUTPUT = F_3)
C_3 CARRY TO RESET
CONTROLLED BY BIT TIME LOGIC
NORMAL ADDER QUALIFYING LOGIC (OUTPUT = F_d)
ACCUMULATOR REGISTER (A_0)
PULSE ACCUMULATOR PULSE HOLD FLIP FLOPS (OUTPUTS = V_1-V_6, T_s)
INCREMENTAL INPUTS (Δ1, Δ2, Δ3, ΔT)
INPUT
Y_{o2}
RFF
Y_{o1}
RFF
Y_0
WA
SIGMATOR LONG LINE (INTEGRAND REGISTER)
8 WORD DELAY
32 WORD DELAY

Figure 4-24. Sigmator, Flow Diagram


INPUT-OUTPUT SELECTOR
PULSE ACCUMULATOR PULSE HOLD FLIP-FLOPS (+ INCREMENTS)
(+) "CARRY" REGISTER
INCREMENTAL ADDER
INCREMENTAL ADDER LOGIC CONTROL
PULSE ACCUMULATOR PULSE HOLD FLIP-FLOPS (− INCREMENTS)
(−) "BORROW" REGISTER
NORMAL PULSE ACCUMULATOR
PULSE CONTROL REGISTER
1 WORD
6 WORDS
EMF
WA
RFF
F1, F2
+ΔVu
+ΔVv
+ΔVw
ΔTi
−ΔVu
−ΔVv
−ΔVw
W1
W2
W5
V1
V3
V5
Ts
V2
V4
V6
C4
C5
Fc
Ft
Ff
K4
K5
K6
x0
Xo1
Yo2



NOTES:

$P_{23} = \overline{T}_5\overline{T}_4\overline{T}_3\overline{T}_2T_1$

$P_{22} = \overline{T}_5\overline{T}_4\overline{T}_3T_2\overline{T}_1$

$P_{21} = \overline{T}_5\overline{T}_4\overline{T}_3T_2T_1$

$P_{20} = \overline{T}_5\overline{T}_4T_3\overline{T}_2$

$P_{19} = \overline{T}_5T_4\overline{T}_3\overline{T}_2\overline{T}_1$

$P_{18} = \overline{T}_5T_4\overline{T}_2T_1$

$P_{13} = T_5\overline{T}_4\overline{T}_3\overline{T}_2\overline{T}_1$

$P_{12} = T_5\overline{T}_4\overline{T}_3\overline{T}_2T_1$

$P_5 = T_5T_4T_2\overline{T}_1$

$P_4 = T_5T_4\overline{T}_3T_2T_1$

$P_1 = \overline{T}_5T_3T_2\overline{T}_1$

$P_0 = T_3T_2T_1$

$\left.\begin{matrix} P_{24}\text{-}P_{20}, \\ P_{17}\text{-}P_{14}, \\ P_{11}\text{-}P_{6}, \\ P_4\text{-}P_1, \\ \overline{P}_0\ \overline{P}_5\ \overline{P}_{13}\ \overline{P}_{19} \end{matrix}\right\} = \overline{P}_0\,\overline{P}_5\,(T_3 + T_2 + T_1 + T_5T_4 + \overline{T}_5\overline{T}_4)$

$\overline{FW} = K_1 + K_2$

$\overline{Sr\,(A_0 \cdot y_0)} = \overline{O}_2 + \overline{O}_3 + S_5 + S_4 + S_3 + S_2 + S_1$

$\phi 1 = \overline{K}_4K_5K_6$

$\phi 2 = K_4\overline{K}_5K_6$

$\phi 3 = K_4K_5K_6$

$\phi 4 = \overline{K}_4\overline{K}_5K_6$

$\phi 5 = \overline{K}_4K_5\overline{K}_6$

$\phi 6 = K_4\overline{K}_5\overline{K}_6$

$\phi 7 = \overline{K}_4\overline{K}_5\overline{K}_6$

$\phi 8 = K_4K_5\overline{K}_6$

$F_v = \phi\overline{4} = K_4 + K_5 + \overline{K}_6$

$\phi\overline{5} = K_4 + \overline{K}_5 + K_6$

$\phi\overline{6} = \overline{K}_4 + K_5 + K_6$

$\phi\overline{7} = K_4 + K_6$

Figure 4-25. Sigmator, Logic Diagram

4-75. Phase Control Register. Flip-flops K_4, K_5, and K_6 and their associated logic constitute the phase control register. The flip-flops are program-controlled by information stored in the S_{ot} track. Flip-flops T_4 and T_5, as used in phase control, are time-shared with bit counter usage in defining pulse times.

4-76. Serial Full Adder (Two-Input Adder). The serial full adder accepts the output of emitter followers F_1 and F_2 and adds serially by using "carry" flip-flop C_3. The output of the adder is fed through emitter follower F_3 to the integrand register.

4-77. Integrand Register (Sigmator Long Line). The integrand register, or sigmator long line, consists of a 32-word section of a track on the drum. One write head y_0 and two read heads Y_{o1} and Y_{o2} provide for input and output of the register. The output of emitter follower F_d qualifies the recording of either the output of the serial full adder or the output of the accumulator register A_0. Read heads Y_{o1} and Y_{o2}, spaced 8 and 32 words behind y_0, apply the output of the integrand register to either the serial adder input selector or the accumulator register A_0.

4-78. Input-Output Selector. The input-output selector consists of flip-flops W_1, W_2 and W_5 and associated logic control. They are used to qualify the various inputs and outputs within the sigmator. An example of their use is the selection of an output from one of the read heads of the pulse accumulator.

4-79. Functional Cycle

The functional cycle of the sigmator consists of eight phases which are program controlled. All operations of the sigmator are accomplished during the eight phases. Figure 4-25 defines the logic for each operation and includes the logic for the phase control register. The phase control flip-flops K_4, K_5, and K_6 are set at P_0 by T_4, T_5, and S_{ot} with additional control from the integration control bit which is stored in X_{o1} at P_0. Flip-flops T_4 and T_5 are set by S_{ot} codes at P_2 and P_1 respectively. The input-output selection flip-flops, W_1 and W_2, which also select the read head of the pulse accumulator (short line), are controlled by information from S_{ot} at P_{20} and P_{21}. The read heads of the pulse accumulator are spaced so that the appropriate group of pulse inputs in the pulse accumulator is selected and read by X_{o2} when that corresponding function is being read by the integrand register read head Y_{o2}. The integrand register, being 32 words in length, cycles twice for each drum revolution. Thus the pulse accumulator is read, or cleared, each half revolution of the drum.

The following list of the phases and their functions is followed by a description of the individual phases.

PHASE	PHASE CONTROL REGISTER	FUNCTION
1	$\overline{K}_4\ K_5\ K_6$	Accept summation of random pulses from short line via X_{o2}. ($Y_{o2} + X_{o2} = y_0$).
2	$K_4\ \overline{K}_5\ K_6$	Integrate a function with respect to time. ($Y_{o2} + Y_{o1}\ \Delta T_{200} = y_0$).
3	$K_4\ K_5\ K_6$	Process any two-word length integral and propagate any carry required. ($Y_{o2} + Y_{o1}\ \Delta T_{200} + C_3 = y_0$).
4	$\overline{K}_4\ \overline{K}_5\ K_6$	Shift data link word to left. ($Y_{o2} + Y_{o2} = y_0$).
5	$\overline{K}_4\ K_5\ \overline{K}_6$	Add count down increments at rate of 1300/sec. during an 8-word recirculation loop. ($Y_{o1} + \Delta T_{800} = y_0$) $T_{800} = X_{o1}\ P_0$
6	$K_4\ \overline{K}_5\ \overline{K}_6$	Recirculate 32-word information unchanged. ($Y_{o2} + 0 = y_0$).
7	$\overline{K}_4\ \overline{K}_5\ \overline{K}_6$	Accept M_n converter scan. ($M_n + 0 = y_0$).
8	$K_4\ K_5\ \overline{K}_6$	Allow propagation of carry, 32-word recirculating loop. ($Y_{o2} + C_3 = y_0$).

4-80. Phase Description.

4-81. Phase 1. During phase 1 the accumulated increments in the pulse accumulator (short line) are added into the integrand register (long line). The heads of the pulse accumulator are so spaced that while the summation of the increments of a function is in the read flip-flop X_{o2}, the corresponding function is in Y_{o2}. Both X_{o2} and Y_{o2} feed the serial adder during phase 1. The output of the serial adder F_3 supplies the write amplifier y_0 of the integrand register. A phase 1 exists for each set of pulse inputs during each 1/2 drum revolution. The sign digit of the increment is held in X_{o2} for each remaining time of phase 1.

4-82. Phase 2. Integration of a selected function takes place in phase 2. The function to be integrated may be one that has been programmed directly into the integrand register or one whose random pulse inputs are being accumulated. In the latter case the summation of the random pulses appears in the integrand register. The function to be integrated appears under the Y_{o1} head of the integrand register while the previous summation appears at the Y_{o2} head. The function is integrated relative to real time at a nominal rate of 200 integration cycles per second. The time control increment is read by the read head of X_{o1} at P_0 for each integration.

4-83. Phase 3. When the summation of some function requires two words to hold the significant information, phase 3 is used. For example, assume a function "F" which varies over a range such that the integral "f" requires two words. The sign digit must be retained during the second word to be added to the most significant half of "F". The X_{o2} flip-flop is set in the sign digit state of the first word at P_0, phase two. During phase 3, the output of X_{o2} is disqualified so the sign digit can be held for all of phase 3. Also during phase 3, any carries to be propagated from phase 2 are held in flip-flop C_3 of the serial adder.

4-84. Phase 4. To provide data link information, the word to be transmitted must be shifted one bit to the left at each 32 word cycle of the sigmator so that at each cycle a different bit of the word is detected. The shift is accomplished during phase 4 by adding the word from Y_{o2} to itself in the serial adder each time it is read.

During the first compute cycle the sign digit is detected at P_0. Then the next most significant digit is detected during the next cycle, and the process is repeated until the least significant digit is detected at P_0 during the last of the twenty-five compute cycles.

4-85. Phase 5. Phase 5 provides an eight word recirculation loop. The loop is used in counting down some value to zero with respect to real time. The eight word loop recirculates at a nominal rate of 800 times per second, because the Y_{o1} read head is spaced eight words after the write head y_0 into which Y_{o1} feeds. The count down is accomplished by adding the time increments ΔT_{800} through the use of the carry flip-flop C_3. The summation is compared with the value of the function to be counted down which is entered in the integrand register. As the difference becomes zero, the sign bit is changed and is recorded at P_0 which indicates the end of the phase.

4-86. Phase 6. Some functions, such as steering and torquing, stored on the integrand register require a 32 word recirculation without change. Phase 6 provides such recirculation by feeding y_0 directly from Y_{o2} spaced 32 words away. A programmed phase 2 or phase 3 is changed to phase 6 if the 200 cycle integration control bit indicates a deletion of integration.

4-87. Phase 7. During phase 7 certain analog input functions are accepted into the serial adder from the M_n converter scan.

4-88. Phase 8. This phase is used in the case of a double-length function such as real time summation when a carry propagation is required. Flip-flop C_3 provides a +1 count for each cycle to accomplish the propagation. The phase is a 32 word recirculating loop. Phase 8 occurs only after phase 1.

The integrand register is also used to store information in and to receive information from the accumulator register A_0. During FW phase of a normal store order, the accumulator register may feed into the integrand register via y_0. Outputs from the integrand register may feed into the accumulator register through Y_{o1} or Y_{o2} during certain modified store orders. These orders are specified in the section on orders in this chapter.

4-89. LOGIC EQUATIONS

K_1, K_2, K_3 PHASE CONTROL FLIP-FLOPS

$$k_1 = R_3 + P_0 \left\{ \overline{K}_2 \left[\overline{O}_1 O_2 + O_1 O_3 (S_5 + S_4 S_3) + O_1 \overline{O}_2 O_3 \right] + \overline{J} + K_2 K_c \left[\overline{K}_3 + \overline{O}_1 O_2\ O_3) \right] \right\}$$

$$\overline{k}_1 = J\ P_0\ K_2\ (F + \overline{K}_3)\ R_4 + S_{0t}\ R_1\ \overline{T}_5\ T_4\ \overline{T}_2\ T_1$$

$$k_2 = R_3 + P_0 \left\{ K_1 \overline{K}_3 + K_1 \overline{O}_1 O_3 \overline{S}_4 \overline{S}_3 \overline{S}_2 \overline{S}_1 + \overline{J} + \overline{K}_1 \left[O_1 \overline{O}_3 + O_1 O_2 \overline{S}_5 \cdot (\overline{S}_4 + \overline{S}_3) + \overline{O}_1 \overline{O}_2 \right] \right\}$$

$$\overline{k}_2 = J\ P_0\ \overline{K}_1\ K_3\ K_c\ (O_1 + \overline{O}_2 + \overline{O}_3)$$

$$k_3 = R_3 + P_0 \left[\overline{J} + K_1 K_2 + \overline{K}_1 \overline{K}_2 O_1 O_3 (S_5 + S_4 S_3 + \overline{O}_2) \right] + \overline{K}_1 \overline{K}_2 \overline{O}_1 O_2 (A_0 + P_0)$$

$$\overline{k}_3 = \overline{K}_1\ \overline{K}_2\ \overline{O}_1\ O_2 \overline{A}_0 \overline{P}_0 + A_1 P_{24} \overline{K}_1 \overline{O}_1 O_2 O_3 + P_0\ J\ (K_1 \overline{K}_2 S_5 + K_1 K_2\ Md\ F + \overline{K}_1 K_2 K_c)$$

Computer Phases

PHASE	State of the K_1, K_2, and K_3 FLIP-FLOPS
W_α	$\overline{K}_1\ K_2\ \overline{K}_3$
I_R	$K_1\ K_2\ \overline{K}_3$
W_β	$\overline{K}_1\ K_2\ K_3$
F_W	$\overline{K}_1\ \overline{K}_2$
A_W	$K_1\ \overline{K}_2\ K_3$
S_P	$K_1\ K_2\ K_3$
L_W	$K_1\ \overline{K}_2\ \overline{K}_3$

K_4, K_5, K_6 SIGMATOR CONTROL FLIP-FLOPS

$$k_4 = T_4 P_0$$

$$\overline{k}_4 = \overline{T}_4 P_0$$

$$k_5 = T_5 P_0 (X_{01} + \overline{T}_4 + \overline{S}_{0t})$$

$$\overline{k}_5 = \overline{T}_5 P_0 + T_4 S_{0t} \overline{X_{01}} P_0$$

$$k_6 = S_{0t} P_0 (X_{01} + \overline{T}_4)$$

$$\overline{k}_6 = \overline{S}_{0t} P_0 + \overline{X}_{01} T_4 P_0$$

SIGMATOR PHASES

Phase	State Of K_4, K_5, K_6 F.F.'s	Operation Performed	
1	$\overline{K}_4 K_5 K_6$	$\dot{X} + \Delta\dot{X} = \dot{X}n$	$Y_{02} + X_{02} = y_o$
2	$K_4 \overline{K}_5 K_6$	$X + \dot{X} = Xn$(integrate)	$Y_{02} + Y_{01} = y_o$
3	$K_4 K_5 K_6$	Integration with sign propogation	$Y_{02} + Y_{01} + (C_3) = y_o$
4	$\overline{K}_4 \overline{K}_5 K_6$	Left shift for data link	$Y_{02} + Y_{02} = y_o$
5	$\overline{K}_4 K_5 \overline{K_6}$	Main engine cutoff countdown	$Y_{01} + \Delta T_{800} = y_o$
6	$K_4 \overline{K_5} \overline{K_6}$	Recirculate	$Y_{02} + 0 = y_o$
7	$\overline{K}_4 \overline{K}_5 \overline{K}_6$	Input to Y_0 from A/D Converter Disk	$M_n + 0 = y_o$
8	$K_4 K_5 \overline{K_6}$	Recirculate with Carry Propogation	$Y_{02} + (C_3) = y_o$

A_1, D_1, - Accumulator Input FF's

$$\begin{aligned} a_1 = {} & G A_0 \left\{ K_2 + K_1O_2 + \overline{K}_1O_1 \left[F_j + S_3M_n + S_1S_3 \right] \right\} + \overline{K}_1\overline{K}_2O_2O_3\overline{S}_5S_4\overline{S}_3(Y_{02}\overline{S}_2 \\ & + Y_{01}S_2S_1 + M_n\overline{S}_1) + K_1\overline{K}_2\overline{O}_2\overline{K}_3R_0 + K_1\overline{K}_2K_3\overline{O}_2S_1\overline{P}_{24} + \overline{G} E_3 \end{aligned}$$

$$\begin{aligned} \overline{a}_1 = {} & G \overline{A}_0 \left[K_2 + K_1O_2 + \overline{K}_1O_1 (F_j + S_3) \right] + \overline{K}_1\overline{K}_2O_2O_3\overline{S}_5S_4 \left[\overline{M}_n\overline{S}_1 + \overline{S}_3(\overline{Y}_{02}\overline{S_2} \right. \\ & \left. + \overline{Y}_{01}S_2S_1) \right] + K_1\overline{K}_2\overline{O}_2\overline{K}_3\overline{R}_0 + K_1\overline{K}_2K_3\overline{O}_2\overline{S}_1 + K_1\overline{K}_2K_3\overline{O}_2P_{24} + \overline{K}_1\overline{K}_2\overline{O}_1 + \overline{G}\,\overline{E}_3 \end{aligned}$$

$$\begin{aligned} d_1 = {} & \overline{K}_1\overline{K}_2M_{0y} (\overline{O}_3\overline{O}_2 + \overline{O}_3K_c + A_0\overline{O}_1O_3) + K_1\overline{K}_2\overline{O}_2K_3S_2 + K_1\overline{K}_2\overline{O}_2\overline{K}_3R_0 \\ & + K_1\overline{K}_2K_3O_1O_2S_2A_0 + K_1\overline{K}_2\overline{O}_3\overline{K}_cD_0\overline{P}_0 \end{aligned}$$

$$\begin{aligned} \overline{d}_1 = {} & K_2 + \overline{K}_1\overline{M_{0y}} + \overline{A}_0O_3 (O_2 + \overline{K}_1) + K_1 \left[\overline{O}_3\overline{D}_0\overline{P}_0 + K_c\overline{O}_3 + \overline{O}_2 (K_3\overline{S}_2 + \overline{K}_3P_0 \right. \\ & \left. + \overline{K}_3\overline{R}_0P_{24}) \right] \end{aligned}$$

C_A Carry FF

$$c_A = \left[\overline{K}_1\overline{K}_2M_{0y}O_1\overline{O}_2O_3P_0 + \overline{K}_1\overline{K}_2K_cA_1\overline{D}_1O_1 + K_c\overline{A}_1D_1 (K_1 + \overline{O}_1) + \overline{K}_cA_1D_1 \right] \overline{P_{24}}$$

$$\overline{c_A} = P_{24} + \overline{K}_1\overline{K}_2O_1K_c\overline{A}_1D_1 + K_cA_1\overline{D_1} (K_1 + \overline{O}_1) + \overline{K}_c\overline{A}_1\overline{D}_1$$

Output Control, Incremental Input Head Select, Time Control Storage, and Mox Recirculation Control FF's

$w_1 = S_{ot} \overline{T}_5 \overline{T}_4 T_3 \overline{T}_2 \overline{T}_1$

$\overline{w}_1 = \overline{S_{ot}} \overline{T}_5 \overline{T}_4 T_3 \overline{T}_2 \overline{T}_1$

$w_2 = S_{ot} \overline{T}_5 \overline{T}_4 \overline{T}_3 T_2 T_1$

$\overline{w}_2 = \overline{S}_{ot} \overline{T}_5 \overline{T}_4 \overline{T}_3 T_2 T_1$

$w_3 = S_{ot} \overline{T}_5 \overline{T}_4 \overline{T}_3 T_2 \overline{T}_1$

$\overline{w}_3 = \overline{S}_{ot} \overline{T}_5 \overline{T}_4 \overline{T}_3 T_2 \overline{T}_1$

$w_4 = S_{ot} \overline{T}_5 \overline{T}_4 \overline{T}_3 \overline{T}_2$

$\overline{w}_4 = \overline{S}_{ot} \overline{T}_5 \overline{T}_4 \overline{T}_3 \overline{T}_2 T_1$

$w_5 = S_{ot} P_{24} + F_3 \overline{T}_5 \overline{T}_4 \overline{T}_3 (\overline{T}_2 T_1 + T_2 \overline{T}_1) \overline{K}_4 K_5 K_6 W_1 W_2$

$\overline{w}_5 = P_0$

$W_5 W_4 =$ Y_{02} OUTPUT - MoX RECIRCULATES

$W_5 \overline{W}_4$ = Mox OUTPUT - NON RECIRCULATION

$\overline{W}_5$ = RECIRCULATE Mox

W_5 USED TO TIME CONTROL INTEGRATIONS

$W_1 W_2$ - Time Head

$\overline{W}_1 W_2$ $\Delta 3$ Head

$W_1 \overline{W}_2$ Δ_2 Head

$\overline{W}_1 \overline{W}_2$ Δ_1 Head

K_c Coincidence and Adder Control Flip-Flop

$$k_c = \overline{K}_1 K_2 \overline{T}_5 \overline{T}_3 + \overline{K}_1 \overline{K}_3 \overline{O}_3 P_0 + K_1 \overline{K}_2 \overline{O}_3 \overline{S}_3 P_0 + K_1 \overline{K}_2 \overline{O}_2 \overline{F}_a P_{24}$$

$$\overline{k_c} = \overline{K}_1 K_2 P_0 \left(\overline{O_2} + O_3 + \overline{A}_0 \overline{O}_1\right) + \overline{K}_1 \overline{K}_2 K_3 P_0 \overline{O_3} + K_1 \overline{K_2} O_3 F_a P_{24} + K_1 \overline{K_2} S_3 O_2 P_0$$

$$+ \overline{K}_1 K_2 F_i T_5 \left[\overline{T}_4 K_3 (\overline{T}_3 + \overline{T}_2) + T_4 \overline{K3}\right]$$

S_1, S_2, S_3, S_4, S_5 Track Selector FF's

$$s_1 = K_1 \overline{K_2} K_3 A_0 \overline{O_2} + S_2 F_s$$

$$\overline{s_1} = K_1 \overline{K_2} K_3 \overline{A_0} \overline{O_2} + \overline{S_2} F_s$$

$$s_2 = D_0 \left(\overline{K}_1 \overline{K_2} P_0 \overline{O_2} + K_1 \overline{K_2} K_3 \overline{O_2} + K_1 \overline{K_2} K_3 \overline{O_3}\right) + S_3 F_s$$

$$\overline{s_2} = \overline{D_0} \left(\overline{K}_1 \overline{K_2} P_0 \overline{O_2} + K_1 \overline{K_2} K_3 \overline{O_2} + K_1 \overline{K_2} K_3 \overline{O_3}\right) + \overline{S_3} F_s$$

$$s_3 = K_1 \overline{K_2} K_3 R_0 K_c \overline{O_2} + \overline{K}_1 \overline{K}_2 P_0 \overline{O_2} + S_4 \left(F_s + K_1 \overline{K_2} K_3 R_0 \overline{O_2} + K_1 \overline{K_2} K_3 \overline{O_3}\right)$$

$$\overline{s_3} = K_1 \overline{K_2} K_3 \overline{R_0} \overline{O_2} + \overline{S_4} \left(F_s + K_1 \overline{K_2} K_3 \overline{O_3} + K_1 \overline{K_2} K_3 \overline{O_2} \overline{K_c}\right)$$

$$s_4 = R_0 K_1 \overline{K_2} K_3 \overline{O_3} + K_1 \overline{K_2} K_3 \overline{O_2} R_0 + S_5 F_s$$

$$\overline{s_4} = \overline{R_0}\, K_1 \overline{K_2} K_3 \overline{O_3} + K_1 \overline{K_2} K_3 \overline{O_2} \overline{R_0} P_{24} + \overline{K}_1 \overline{K}_2 P_0 \overline{O_3} + \overline{S_5}\, F_s$$

$$s_5 = K_1 \overline{K_2} K_3 \overline{T_5} \overline{T_3} + I_0 F_s$$

$$\overline{s_5} = K_1 \overline{K_2} K_3 T_5 T_4 F_i + \overline{I_0} F_s$$

Ordinary Store and Modified Store Commands

S_5	S_4	S_3	S_2	S_1	Fw		Aw	
0	0	0	0	0	A_0 stored in Y_0			
0	0	0	0	1	A_0 stored in $M_{o v}$			
0	0	0	1	0	A_0 stored in M_{ox}			
0	0	0	1	1	A_0 stored in M_{ov} and M_{ox}			
0	0	1	0	0	A_0 stored in M_{oz}			
0	0	1	0	1	A_0 stored in M_{ov} and M_{oz}			
0	0	1	1	0	A_0 stored in M_{ox} and M_{oz}			
0	0	1	1	1	A_0 stored in M_{ov} and M_{ox} and M_{oz}			
0	1	0	0	0	Not used as a command			
0	1	0	0	1	Y_{02} read into A_0			
0	1	0	1	0	M_n read into A_0			
0	1	0	1	1	Y_{01} read into A_0			
0	1	1	0	0	Extract A_0 with M_n	Goes to Sr Aw	Right shift during Aw	Recirculate during Lw
0	1	1	0	1	Recirculate			
0	1	1	1	0	Extract A_0 with M_n		Left shift during Aw	
0	1	1	1	1	Recirculate			
1	0	0	0	0	Recirculate during Fw. Then go to Tc Aw		STOP	
1	0	0	0	1			RELAY SET COMMANDS	
1	0	0	1	0				
1	0	0	1	1				
1	0	1	0	0				
1	0	1	0	1				
1	0	1	1	0				
1	0	1	1	1				
1	1	0	0	0				
1	1	0	0	1				
1	1	0	1	0				
1	1	0	1	1				
1	1	1	0	0				
1	1	1	0	1				
1	1	1	1	0				
1	1	1	1	1				

P	T_5	T_4	T_3	T_2	T_1	
24	0	0	0	0	0	α_T
23	0	0	0	0	1	
22	0	0	0	1	0	
21	0	0	0	1	1	
20	0	0	1	0	0	
19	0	1	0	0	0	β_T
18	0	1	0	0	1	
17	0	1	0	1	0	
16	0	1	0	1	1	
15	0	1	1	0	0	
14	1	0	1	0	1	β_S
13	1	0	0	0	0	
12	1	0	0	0	1	
11	1	0	0	1	0	
10	1	0	0	1	1	
9	1	0	1	0	0	
8	1	1	1	0	1	α_S
7	1	1	0	0	0	
6	1	1	0	0	1	
5	1	1	0	1	0	
4	1	1	0	1	1	
3	1	1	1	0	0	
2	1	0	1	1	0	O_P
1	0	x	1	1	0	
0	x	x	1	1	1	

$$t_5 = S_{0t}\,(T_3 T_4 \overline{T_2} + T_2 \overline{T_1} T_3)$$

$$\overline{t_5} = T_3 T_2$$

$$t_4 = T_3 \overline{T_1} \left[\overline{T_2} + T_5 S_{ot} \right]$$

$$\overline{t_4} = T_3\, \overline{T_2}\,\overline{T_1} + P_0$$

$$t_3 = T_2 T_1$$

$$\overline{t_3} = T_1 + \overline{T}_5 \overline{T}_4 \overline{T}_2$$

$$t_2 = \overline{T}_3 T_1 + T_5 T_4 T_3 \overline{T_1}$$

$$\overline{t_2} = T_1$$

$$t_1 = \overline{T_3} + \overline{T}_5 T_4 S_{ot} + T_5 \overline{T_4}\,\overline{T_2} + \overline{T}_5 T_2$$

$$\overline{t_1} = 1$$

m_n - Converter Scan F. F.

$$\begin{aligned} m_n = \; & T_5\overline{T_4}\left[B_8^* T_3\overline{T}_2\overline{T}_1 + B_{13}^* T_3\overline{T}_2T_1 + B_1^* T_3T_2\overline{T}_1 + B_9^* \overline{T}_3T_2T_1 + B_{10}^* \overline{T}_3T_2\overline{T}_1\right. \\ & \left. + B_{11}^* \overline{T}_3\overline{T}_2T_1 + B_{12}^* \overline{T}_3\overline{T}_2\overline{T}_1\right] + T_5T_4\left[B_2^* T_3\overline{T}_2\overline{T}_1 + B_7^* T_3\overline{T}_2T_1\right. \\ & \left. + B_3^* \overline{T}_3T_2T_1 + B_4^* \overline{T}_3T_2\overline{T}_1 + B_5^* \overline{T}_3\overline{T}_2T_1 + B_6^* \overline{T}_3\overline{T}_2\overline{T}_1\right] \\ & + \overline{T_5}\left\{ T_4(B_{14}^* T_3\overline{T}_2\overline{T}_1 + B_{15}^* \overline{T}_3T_2T_1) + B_0^* T_3T_2\overline{T_1}\right\} \end{aligned}$$

$$\begin{aligned} \overline{m}_n = \; & T_5\overline{T}_4\left[B_8T_3\overline{T}_2\overline{T}_1 + B_{13}T_3\overline{T}_2T_1 + B_1T_3T_2\overline{T}_1 + B_9\overline{T}_3T_2T_1 + B_{10}\overline{T}_3T_2\overline{T}_1\right. \\ & \left. + B_{11}\overline{T}_3\overline{T}_2T_1 + B_{12}\overline{T}_3\overline{T}_2\overline{T}_1\right] + T_5T_4\left[B_2 T_3\overline{T}_2\overline{T}_1 + B_7 T_3\overline{T}_2T_1 + B_3\overline{T}_3T_2T_1\right. \\ & \left. + B_4\overline{T}_3T_2\overline{T}_1 + B_5\overline{T}_3\overline{T}_2T_1 + B_6\overline{T}_3\overline{T}_2\overline{T}_1\right] + \overline{T}_5(B_0T_3T_2\overline{T}_1 + B_{14}T_4T_3\overline{T}_2\overline{T}_1) + P_0 \end{aligned}$$

A_0, D_0, R_0, I_0 Recirculation Registers

$$a_0 = \overline{E_f}\,(F_a\overline{D_1} + \overline{F_a}D_1) + E_f\,F$$
$$\overline{a_0} = \overline{E_f}\,(F_aD_1 + \overline{F}_a\overline{D}_1) + E_f\,L$$

$$d_0 = K_1\overline{K_2}\,(S_2O_3 + S_2P_0 + \overline{O}_3D_0\overline{P_0}) + \overline{K}_1M_{oy} + E_fI_0$$
$$\overline{d_0} = K_1\overline{K_2}\,(\overline{S}_2O_3 + \overline{S}_2P_0 + \overline{O}_3\overline{D}_0\overline{P}_0) + \overline{K}_1\overline{M_{oy}} + E_f\,\overline{I_0}$$

$$r_0 = K_1\overline{K_2}\,(S_3\overline{P_{24}} + S_3O_3 + O_3\overline{S}_4R_0) + \overline{K}_1K_3\overline{O_3}$$
$$\overline{r_0} = K_1\overline{K_2}\overline{O_3}\,(P_{24} + \overline{S}_3) + \overline{K}_1\overline{K}_2\overline{K}_3 + \overline{K}_2O_3\overline{S}_3\,(S_4 + \overline{R_0})$$

$$i_0 = R_3 + K_1\overline{K_3}M_{oy} + \overline{E_f}\,\overline{L}\,(\overline{K}_1 + K_3)\,I_0 + \overline{E_f}E_3L$$
$$\overline{i_0} = K_1\overline{K_3}\,\overline{M_{oy}} + \overline{E_f}\,\overline{L}\,(\overline{K}_1 + \overline{K_3R_3})\,\overline{I_0} + \overline{E_f}\,L\,\overline{E_3}$$

x_0, Short Line Sigmator Write Amplifier

$$x_0 = \overline{E_f}\Big\{\overline{P_5}\big[\overline{P_0}\, F_c\, (\overline{K_4}\,\overline{K_5}K_6 + \overline{X_{01}}) + X_{01}\overline{F_c}F_v\big] + P_0 \big[(W_5 + \overline{X_{01}}).$$

$$\cdot\, \overline{K_4}K_5K_6W_1W_2 + X_{01}\, F_f\big] + P_5 \big[(\overline{K_4}K_5\overline{K_6} + \overline{X_{01}})\, T_s + X_{01}\overline{T_s}\, (K_4 + \overline{K_5} + K_6)\big]\Big\} + E_fF$$

$$\overline{x_0} = \overline{E_f}\Big\{\overline{P_5}\,\overline{P_0}\, \big[F_cX_{01}F_v + \overline{F_c}\, (\overline{X_{01}} + \overline{K_4}\,\overline{K_5}K_6)\big] + P_0 \big[\overline{X_{01}}F_f + X_{01}\overline{W_5}\,\overline{K_4}K_5K_6W_1W_2\big]$$

$$+ P_5 \big[(\overline{X_{01}} + \overline{K_4}K_5\overline{K_6})\, \overline{T_s} + T_sX_{01}\, (K_4 + \overline{K_5} + K_6)\big]\Big\} + E_f\, L$$

y_0 Long Line Sigmator Write Amplifier

$$y_0 = \overline{E_f}F_3 + A_0\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_4}\,\overline{S_3}\,\overline{S_2}\,\overline{S_1}\,\overline{S_5} + E_fF$$

$$y_0 = F_d\, (\overline{F_1}\,\overline{F_2}\,\overline{C_3} + \overline{F_1}F_2C_3 + F_1\overline{F_2}C_3 + F_1F_2\overline{C_3})\, \overline{E_f} + \overline{A_0}\,\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_4}\,\overline{S_3}\,\overline{S_2}\,\overline{S_1}S_5 + E_fL$$

m_{ov}, m_{ov}, m_{oz} Write Amplifiers

$$m_{\mathbf{ox}} = A_0S_2\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_5}\,\overline{S_4} + \big[M_{ox}\, F_h\, (\overline{W_5} + W_4)\big]\, \overline{E_f}$$

$$\overline{m_{ox}} = \overline{A_0}S_2\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_5}\,\overline{S_4} + \overline{M}_{ox}\, F_h\, (\overline{W_5} + W_4) + E_f$$

$$m_{ov} = A_0S_1\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_5}\,\overline{S_4}$$

$$\overline{m_{ov}} = \overline{A_0}S_1\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_5}\,\overline{S_4}$$

$$m_{oz} = A_0S_3\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_5}\,\overline{S_4}$$

$$\overline{m_{oz}} = \overline{A_0}S_3\overline{K_1}\,\overline{K_2}O_2O_3\overline{S_5}\,\overline{S_4}$$

Y_{01} and X_{02} Read Amplifier Control Emitter Followers Used to Obtain Sign Propagation

Y_{01} Reads normally when

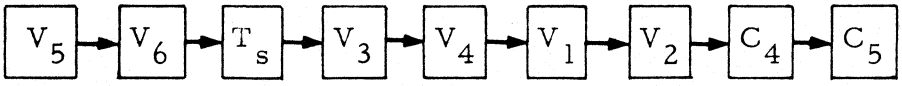
$\overline{K_4} + \overline{K_6} + K_5P_0 + \overline{K_5}\,\overline{P_0} + \overline{K_5}\,\overline{T_5}$ is true

X_{02} Reads normally when

$\overline{T_5}\,\overline{T_4}\, \big[(\overline{W_1} + \overline{W_2})\, \overline{T_2} + \overline{T_3}\, (\overline{W_1} + \overline{W_2} + \overline{T_2} + \overline{T_1})\big] + P_0$ is true

Incremental Input Flip-Flops - Also Used As A Shift Register And A Count-Down Register

Shift Register Order



$$v_1 = (+\Delta V_u)\overline{H} + F_m V_4 T + \overline{F_m}\,\overline{V_2}$$
$$\overline{v}_1 = \left[\overline{H}P_{24}C_4 + F_m\overline{V_4}\right]T + \overline{F_m}\,\overline{V_2}$$

$$v_2 = (-\Delta V_u)\,\overline{H} + F_m V_1 T + \overline{F_m}\,\overline{C_4}$$
$$\overline{v}_2 = \left[P_{24}\overline{H}C_5 + F_m\overline{V_1}\right]T + \overline{F_m}\,\overline{C_4}$$

$$v_3 = (+\Delta V_v)\,\overline{H} + F_m T_s T + \overline{F_m}\,\overline{V_4}$$
$$\overline{v}_3 = \left[\overline{H}C_4\overline{T}_5T_4\overline{T}_2T_1 + F_m\overline{T_s}\right]T + \overline{F_m}\,\overline{V_4}$$

$$v_4 = (-\Delta V_v)\,\overline{H} + F_m V_3 T + \overline{F_m}\,\overline{V_1}$$
$$\overline{v}_4 = \left[\overline{H}C_5\overline{T}_5T_4\overline{T}_2T_1 + F_m\overline{V_3}\right]T + \overline{F_m}\,\overline{V_1}$$

$$v_5 = (+\Delta V_w)\,\overline{H} + F_m A_1 T$$
$$\overline{v}_5 = \left[T_5\overline{T}_4\overline{T}_3\overline{T}_2T_1C_4\overline{H} + F_m\overline{A_1}\right]T + \overline{F_m}\,\overline{V_6}$$

$$v_6 = (-\Delta V_w)\,\overline{H} + F_m V_5 T + \overline{F_m}\,\overline{T_s}$$
$$\overline{v}_6 = \left[T_5\overline{T}_4\overline{T}_3\overline{T}_2T_1C_5\overline{H} + F_m\overline{V_5}\right]T + \overline{F_m}\,\overline{T_s}$$

$$t_s = \Delta t_1\,\overline{H} + F_m V_6 T + \overline{F_m}\,\overline{V_3}$$
$$\overline{t_s} = \left[T_5T_4\overline{T}_3T_2T_1C_4\overline{H}\,\overline{C}_5 + F_m\overline{V_6}\right]T + \overline{F_m}\,\overline{V_3}$$

$$\begin{aligned} c_4 = &\left\{\left[V_1P_0 + (V_3\overline{T}_5T_4 + V_5T_5\overline{T}_4)\,\overline{T}_3\overline{T}_2\overline{T}_1 + T_5T_4T_2\overline{T_1}\,(T_s + X_{01})\right]\overline{H} + F_m V_2\right\}T \\ &+ \overline{F_m}\,\overline{C_5} \end{aligned}$$

$$\begin{aligned}\overline{c_4} = \Big\{ \Big[& \overline{V_1}P_0 + (\overline{V_3}\,\overline{T_5}T_4 + \overline{V_5}T_5\overline{T_4})\,\overline{T_3}\,\overline{T_2}\,\overline{T_1} + \overline{T_s}\,\overline{X_{01}}T_5T_4T_2\overline{T_1} + \overline{K_4}\,\overline{K_5}K_6\overline{C_5} \\ & + X_{01}\overline{C_5}F_t \Big]\,\overline{H} + F_m\overline{V_2} \Big\}\,T + \overline{F_m}\,\overline{C_5}\end{aligned}$$

$$\begin{aligned}c_5 = \Big\{ \Big[& V_2P_0 + (V_4\overline{T_5}T_4 + V_6T_5\overline{T_4})\,\overline{T_3}\,\overline{T_2}\,\overline{T_1} + X_{01}\,\overline{T_s}T_5T_2\overline{T_1}T_4 \Big]\overline{H} + F_mC_4 \Big\}\,T \\ & + \overline{F_m}\Delta t_2\end{aligned}$$

$$\begin{aligned}\overline{c_5} = \Big\{ \Big[& \overline{V_2}P_0 + (\overline{V_4}\,\overline{T_5}T_4 + \overline{V_6}T_5\overline{T_4})\,\overline{T_3}\,\overline{T_2}\,\overline{T_1} + T_5T_4T_2\overline{T_1}\,(\,\overline{X_{01}} + T_s) + X_{01}\,\overline{C_4}F_tF_v \Big]\,\overline{H} \\ & + F_m\overline{C_4} \Big\}\,T + \overline{F_m}\Delta t_2\end{aligned}$$

F_i, F_a, F_b, F_s, and Bit Time Emitter Followers

$$f_i = I_0\overline{S_{0t}} + \overline{I_0}S_{0t}$$

$$f_a = C_A\overline{A_1} + \overline{C_A}\ \Big[A_1F_b + A_0O_1O_2K_1\overline{K_2}K_3\overline{S_2}\,\overline{P_{24}}\Big]$$

$$\overline{f_a} = C_AA_1 + \overline{C_A}\ \Big[\overline{A_1}F_b + \overline{A_0}O_1O_2K_1\overline{K_2}K_3\overline{S_2}\,\overline{P_{24}}\Big]$$

$$f_b = \overline{O_1} + \overline{O_2} + \overline{K_1} + K_2 + \overline{K_3} + P_{24} + S_2$$

$$f_s = \overline{T_5}\ (\overline{T_3} + \overline{T_2})\ \Big[\overline{K_1}K_2\ (K_3 + \overline{T_4}) + K_1\overline{K_2}\,\overline{K_3}\,\overline{T_4}\Big]$$

$$P_0 = T_3T_2T_1$$

$$\overline{P_0} = \overline{T_3} + \overline{T_2} + \overline{T_1}$$

$$P_{24} = \overline{T_5}\,\overline{T_4}\,\overline{T_3}\,\overline{T_2}\,\overline{T_1}$$

$$\overline{P_{24}} = T_5 + T_4 + T_3 + T_2 + T_1$$

$$\overline{P_5} = \overline{T_5} + \overline{T_4} + \overline{T_2} + T_1$$

$$f_t = \overline{P_0}\,\overline{P_5}\ \Big[T_3 + T_2 + T_1 + T_5\ T_4 + \overline{T_5}\,\overline{T_4}\Big]$$

$$f_c = \overline{H}\ (C_4\overline{C_5} + \overline{C_4}C_5)$$

$$\overline{f_c} = C_4C_5 + \overline{C_4}\,\overline{C_5} + H$$

$$f_m = F_g\ \overline{S_4}S_3S_2S_1\ H$$

$$\overline{f_m} = H\ (O_1 + \overline{O_3} + S_4 + \overline{S_3} + \overline{S_2} + \overline{S_1} + \overline{K_1} + K_2 + \overline{K_3} + R_3)$$

Emitter Followers

$$f_d = K_1 + K_2 + \overline{O_2} + \overline{O_3} + S_4 + S_3 + S_2 + S_1 + S_5$$

$$f_f = K_4 + \overline{K_5} + \overline{K_6} + \overline{W_1} + \overline{W_2}$$

$$f_g = \overline{O_1} O_3 K_1 \overline{K_2} K_3$$

$$f_h = K_1 + K_2 + \overline{O_2} + \overline{O_3} + S_5 + S_4 + \overline{S_2}$$

$$f_1 = Y_{02} (K_4 + K_6) + Y_{01} \overline{K_4} K_5 \overline{K_6} + M_n \overline{K_4} \overline{K_5} \overline{K_6}$$
$$\overline{f_1} = \overline{Y_{02}} (K_4 + K_6) + \overline{Y_{01}} \overline{K_4} K_5 \overline{K_6} + \overline{M_n} \overline{K_4} \overline{K_5} \overline{K_6}$$
$$f_2 = Y_{02} \overline{K_4} \overline{K_5} K_6 + Y_{01} K_4 K_6 + X_{02} \overline{K_4} K_5 K_6$$
$$\overline{f_2} = \overline{Y_{02}} \overline{K_4} \overline{K_5} + \overline{Y_{01}} K_4 + \overline{X_{02}} \overline{K_4} K_5 + \overline{K_6}$$

$K_1 \overline{K_2}$

$\overline{K_1} \overline{K_2}$

$$f_j = \overline{O_2} + \overline{O_3} + \overline{S_4} + S_5$$

$$f_3 = (F_1 F_2 C_3 + \overline{F_1} \overline{F_2} C_3 + \overline{F_1} F_2 \overline{C_3} + F_1 \overline{F_2} \overline{C_3}) F_d$$

$$f_v = K_4 + K_5 + \overline{K_6}$$

O_1, O_2, O_3 Order FF's

$$o_1 = (K_1K_2\overline{K}_3 + L)\ O_2$$

$$\overline{o}_1 = (K_1K_2\overline{K}_3 + L)\ \overline{O}_2 + O_2O_3\overline{K}_1\overline{K}_2S_5P_0$$

$$o_2 = (K_1K_2\overline{K}_3 + L)\ O_3$$

$$\overline{o}_2 = (K_1K_2\overline{K}_3 + L)\ \overline{O}_3 + R_3$$

$$o_3 = K_1K_2\overline{K}_3M_{oy} + L\ E_3$$

$$\overline{o}_3 = K_1K_2\overline{K}_3M_{\overline{o}\,\overline{y}} + L\ \overline{E}_3 + R_3$$

	O_1	O_2	O_3	
Clear and add	0	0	0	At
Extract	0	0	1	Ex
Multiply	0	1	0	Mu
Conditional Transfer	0	1	1	Tc
Add	1	0	0	Ad
Divide	1	0	1	Dv
Subtract	1	1	0	Su
Store	1	1	1	Sr

C_3 - Long Line Sigmator Carry Flip-Flop

$$c_3 = F_1F_2(\overline{P}_0 + T_5T_4) + P_{24}\overline{K}_4K_5\overline{K}_6 + P_0T_5\overline{S}_{ot}\overline{T}_4C_4$$

$$\overline{c}_3 = P_0\left[\overline{T}_4(S_{ot} + \overline{C}_4) + \overline{T}_5\right] + \overline{F}_1\overline{F}_2\left[(S_{ot} + \overline{K}_4K_5K_6)T_5T_4 + \overline{P}_0\right]$$

Output Error Determining Logic

$$f_u = W_4(Y_{02}\overline{M}_n + \overline{Y}_{02}M_n) + \overline{W}_4(M_{ox}\overline{M}_n + \overline{M}_{ox}M_n)$$

$$\overline{f}_u = W_4(Y_{02}M_n + \overline{Y}_{02}\overline{M}_n) + \overline{W}_4(M_{ox}M_n + \overline{M}_{ox}\overline{M}_n)$$

$$c_c = F_u$$

$$\overline{c}_c = P_{24}$$

$$c_n = W_4\overline{Y}_{02}M_n + \overline{W}_4\overline{M}_{ox}M_n$$

$$\overline{c}_n = P_{24} + W_4Y_{02}\overline{M}_n + \overline{W}_4M_{ox}\overline{M}_n$$

$$\overline{f}_o = (C_c + F_u)P_0$$

$$\overline{f}_p = (\overline{C}_c + C_n\overline{F}_u + \overline{C}_nF_u)P_0$$

$$\overline{f}_n = (\overline{C}_n\overline{F}_u + C_nF_u)P_0$$

4-90. LIST OF SYMBOLS

SYMBOL	DEFINITION
A_0	Output of Accumulator Register Read Amplifier Flip-Flop
A_1	Output of Accumulator Register Flip-Flop
B_0	Most Significant, Set "0", Converter Brush Signal
B_0^*	Most Significant, Set "1", Converter Brush Signal
B_1 - B_{14}	Set "0" Converter Brush Signals
B_1^* - B_{14}^*	Set "1" Converter Brush Signals
B_{15}^*	Least Significant, Set "1", Converter Brush Signal
C_3	Output of Sigmator Serial Full Adder "Carry" Flip-Flop
C_4	Output of Pulse Accumulator One Increment Adder-Subtractor "Carry" Flip-Flop
C_5	Output of Pulse Accumulator One Increment Adder-Subtractor "Borrow" Flip-Flop
C_a	Output of Serial Adder-Subtractor "Carry" Flip-Flop
C_c	Output of Error (Positive) Scan Flip-Flop
C_n	Output of Error (Negative) Scan Flip-Flop
D_0	Output of Multiplicand-Divisor Read Amplifier Flip-Flop
D_1	Output of Multiplicand-Divisor Register Flip-Flop
$F_1, \overline{F}_1, F_2, \overline{F}_2$	Output of Sigmator Serial Adder Input Selector Emitter Followers
F_3	Output of Sigmator Serial Adder Emitter Follower
$F_a, \overline{F}_a$	Output of Serial Adder Emitter Followers
F_b	Output of Normal Serial Adder Qualifier Emitter Follower
$F_c, \overline{F}_c$	Output of Incremental Adder Input Emitter Followers
F_d	Output of Normal Sigmator Adder Qualifier Emitter Follower
F_f	Output of Normal Pulse Accumulator Control Emitter Follower
F_g	Output of Relay Set Signal Times Emitter Follower
F_h	Output of Normal Recirculation Qualifier Emitter Follower
F_i	Output of Sector Address Comparator Emitter Follower

SYMBOL	DEFINITION
F_j	Output of Normal Accumulator Recirculation Control Emitter Follower
$F_m, \bar{F}_m$	Output of Shift Register Load and Count Down Control Emitter Followers
$\bar{F}_n$	Output of Error Set Emitter Follower
$\bar{F}_o$	Output of Error Set Emitter Follower
$\bar{F}_p$	Output of Error Set Emitter Follower
F_s	Output of Track Selector Gate Emitter Follower
F_t	Output of Incremental Adder Logic Control Emitter Follower
$F_u, \bar{F}_u$	Output of Output Track Selector Emitter Followers
I_0	Instruction Register Read Flip-Flop
K_1	Output of First Control Flip-Flop
K_2	Output of Second Control Flip-Flop
K_3	Output of Third Control Flip-Flop
K_4	Output of First Sigmator Phase Control Flip-Flop
K_5	Output of Second Sigmator Phase Control Flip-Flop
K_6	Output of Third Sigmator Phase Control Flip-Flop
$(\bar{K}_1\bar{K}_2)$	Output of First Word Phase Emitter Follower
$(K_1\bar{K}_2)$	Output of Additional or Last Word Phase Emitter Follower
K_c	Output of Sector Address Coincidence Scan Flip-Flop
M_n	Output of Converter Scan Flip-Flop
M_{ox}	Output of Output Track Read Amplifier Flip-Flop
M_{oy}	Output of Main Memory Read Amplifier Flip-Flop
0_1	Output of First Order Flip-Flop
0_2	Output of Second Order Flip-Flop
0_3	Output of Third Order Flip-Flop
$P_0, \bar{P}_0$	Output of Bit Time Emitter Followers
$\bar{P}_5$	Output of Bit Time Emitter Follower
$P_{24}, \bar{P}_{24}$	Output of Bit Time Emitter Followers
R_0	Multiplier-Quotient Register Read Flip-Flop
S_{ot}	Output of Sector Address Read Amplifier Flip-Flop
S_1	Output of First Track Selector Flip-Flop
S_2	Output of Second Track Selector Flip-Flop
S_3	Output of Third Track Selector Flip-Flop

SYMBOL	DEFINITION
S_4	Output of Fourth Track Selector Flip-Flop
S_5	Output of Fifth Track Selector Flip-Flop
$S_6, \overline{S}_6$	Track Group Selection Control
T	Output of Clock Signal Generator
T_1	Output of First Pulse Counter Flip-Flop
T_2	Output of Second Pulse Counter Flip-Flop
T_3	Output of Third Pulse Counter Flip-Flop
T_4	Output of Fourth Pulse Counter Flip-Flop
T_5	Output of Fifth Pulse Counter Flip-Flop
T_s	Output of Time Standard Flip-Flop
Δt_1	Output of Time Standard No. 1
Δt_2	Output of Time Standard No. 2
V_1	Output of First Pulse Accumulator Pulse Hold Flip-Flop
V_2	Output of Second Pulse Accumulator Pulse Hold Flip-Flop
V_3	Output of Third Pulse Accumulator Pulse Hold Flip-Flop
V_4	Output of Fourth Pulse Accumulator Pulse Hold Flip-Flop
V_5	Output of Fifth Pulse Accumulator Pulse Hold Flip-Flop
V_6	Output of Sixth Pulse Accumulator Pulse Hold Flip-Flop
W_1	Output of First Input-Output Selector Flip-Flop
W_2	Output of Second Input-Output Selector Flip-Flop
W_3	Output of Third Input-Output Selector Flip-Flop
W_4	Output of Fourth Input-Output Selector Flip-Flop
W_5	Output of Fifth Input-Output Selector Flip-Flop
X_{o1}	Output of First Pulse Accumulator Read Amplifier Flip-Flop
X_{o2}	Output of Second Pulse Accumulator Read Amplifier Flip-Flop
Y_{o1}	Output of First Sigmator Read Amplifier Flip-Flop
Y_{o2}	Output of Second Sigmator Read Amplifier Flip-Flop
a_0	Input to Accumulator Register Write Amplifier
a_1	Input to Accumulator Register Flip-Flop
c_3	Input to Sigmator Serial Adder "Carry" Flip-Flop
c_4	Input to Pulse Accumulator One Increment Adder-Subtractor "Carry" Flip-Flop
c_5	Input to Pulse Accumulator One Increment Adder-Subtractor "Borrow" Flip-Flop

SYMBOL	DEFINITION
c_a	Input to Serial Adder-Subtractor "Carry" Flip-Flop
c_c	Input to Error (Positive) Scan Flip-Flop
c_n	Input to Error (Negative) Scan Flip-Flop
d_0	Input to Multiplicand-Divisor Register Write Amplifier
d_1	Input to Multiplicand-Divisor Register Flip-Flop
fr_4	Input to the Y_4 Relay Driver
fr_8	Input to the Y_8 Relay Driver
i_0	Input to Instruction Register Write Amplifier
k_1	Input to First Control Flip-Flop
k_2	Input to Second Control Flip-Flop
k_3	Input to Third Control Flip-Flop
k_4	Input to First Sigmator Phase Control Flip-Flop
k_5	Input to Second Sigmator Phase Control Flip-Flop
k_6	Input to Third Sigmator Phase Control Flip-Flop
k_c	Input to Sector Address Coincidence Scan Flip-Flop
m_n	Input to Converter Scan Flip-Flop
m_{ov}	Input to Temporary Storage Write Amplifier
m_{ox}	Input to Output-Temporary Storage Write Amplifier
m_{oz}	Input to Temporary Storage Write Amplifier
o_1	Input to First Order Flip-Flop
o_2	Input to Second Order Flip-Flop
o_3	Input to Third Order Flip-Flop
r_0	Input to Multiplier-Quotient Register Write Amplifier
s_1	Input to First Track Selector Flip-Flop
s_2	Input to Second Track Selector Flip-Flop
s_3	Input to Third Track Selector Flip-Flop
s_4	Input to Fourth Track Selector Flip-Flop
s_5	Input to Fifth Track Selector Flip-Flop
t_1	Input to First Pulse Counter Flip-Flop
t_2	Input to Second Pulse Counter Flip-Flop
t_3	Input to Third Pulse Counter Flip-Flop
t_4	Input to Fourth Pulse Counter Flip-Flop
t_5	Input to Fifth Pulse Counter Flip-Flop
t_s	Input to Time Standard Flip-Flop

SYMBOL	DEFINITION
v_1	Input to First Pulse Accumulator Pulse Hold Flip-Flop
v_2	Input to Second Pulse Accumulator Pulse Hold Flip-Flop
v_3	Input to Third Pulse Accumulator Pulse Hold Flip-Flop
v_4	Input to Fourth Pulse Accumulator Pulse Hold Flip-Flop
v_5	Input to Fifth Pulse Accumulator Pulse Hold Flip-Flop
v_6	Input to Sixth Pulse Accumulator Pulse Hold Flip-Flop
w_1	Input to First Input-Output Selector Flip-Flop
w_2	Input to Second Input-Output Selector Flip-Flop
w_3	Input to Third Input-Output Selector Flip-Flop
w_4	Input to Fourth Input-Output Selector Flip-Flop
w_5	Input to Fifth Input-Output Selector Flip-Flop
x_0	Input to Pulse Accumulator Write Amplifier
y_0	Input to Sigmator Write Amplifier

NOTE: Inputs to Emitter Followers are specified by a lower case letter followed by the same subscript that the output has.
Example - fd is the input that produces the Fd output - the same is true of relay drivers.

Section V
PROGRAMMING

5-1. INTRODUCTION

Programming is, in effect, a translation of data—usually mathematical—into information that will be accepted and operated upon by the computer. This section provides a brief explanation of the methods and codes used in translating problems into the formats acceptable for storage in the main memory, the control states which occur, instruction and data words, and the orders normally used in programming.

5-2. MAIN MEMORY

The main memory, as shown in Figure 5-1, is composed of 2368 addresses of permanent storage, 192 addresses of temporary storage, and 384 addresses used for special purposes. The total of 2944 addresses is dispersed in 46 columns (tracks) of 64 rows (sectors) each. The full capacity (25 binary bits) of either instruction or data words may be stored in any sector of the memory until required by the program.

5.3 CONTROL STATES

The three control flip-flops define six control states which enable the computer to transfer information in and out of the memory and do the various arithmetic operations. These states are as follows:

a. Wait α (W α). This state defines the period of time during which the computer is searching for the α sector address (Sα). Since the memory location having the α address contains the next instruction to be used, this control state initiates a search for the next instruction. An "instruction read-in" state (IR) always follows Wait α .

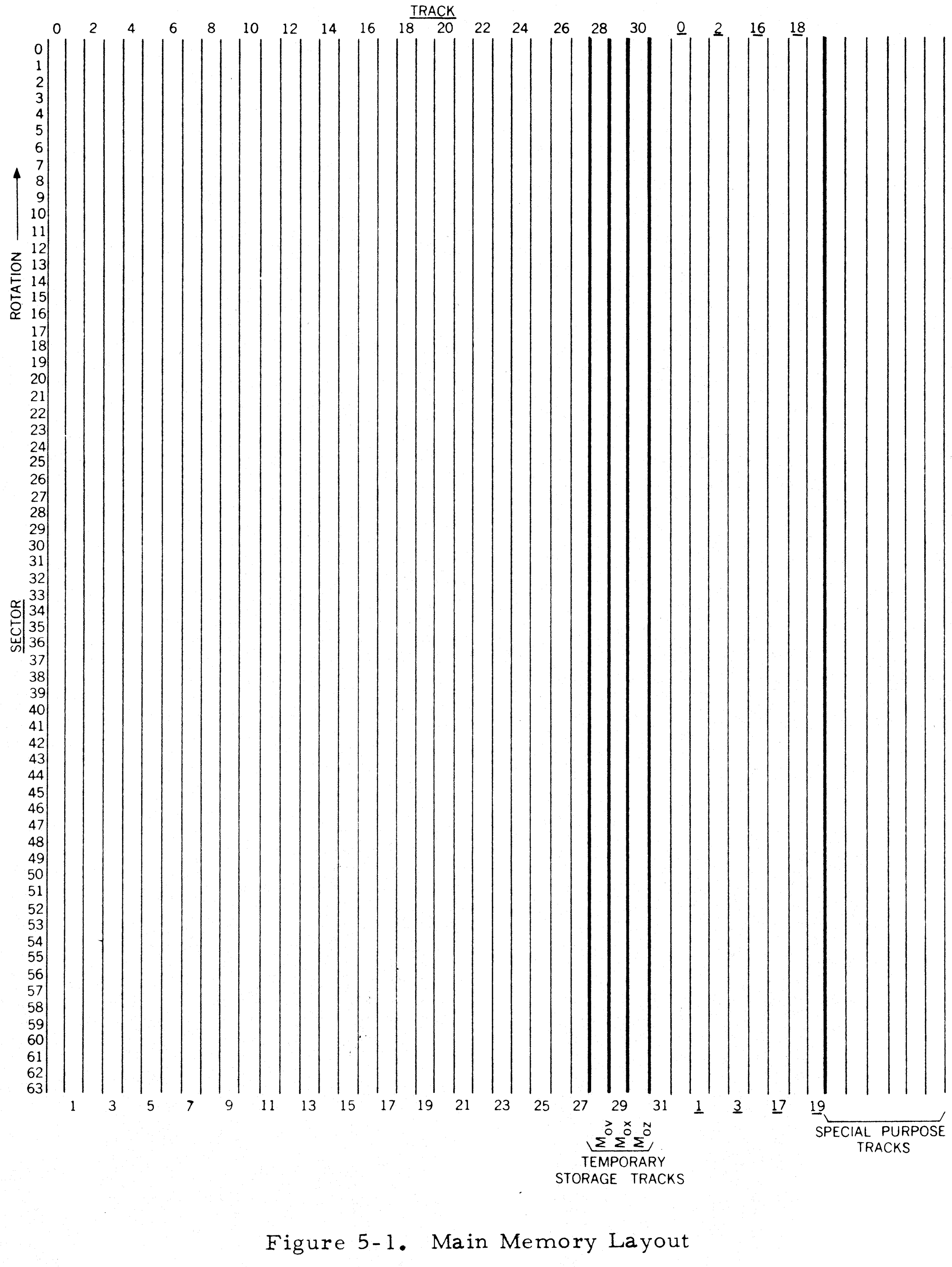


Figure 5-1. Main Memory Layout

b. Instruction Read-In (IR). This state is one word in duration and defines the period in which an instruction is copied from the memory into the recirculating instruction register (Io). The IR state changes to wait β {Wβ) after one word, but Wβ may proceed immediately to the next Wα if the next word to be taken from memory is also an instruction.

c. Wait β (Wβ). This wait state also defines a search; in this case, for the β sector address (Sβ). Since the memory location having the β address may contain either an instruction or an operand, the next state may be either an "instruction read-in" (IR) or a "first word" of operation (FW).

d. First Word (FW). This state defines the first word of operation in the arithmetic unit of the computer. The duration of FW is four word times. The FW state may change to Wα if the arithmetic operation was only one word in duration, or it may change to the "additional words" state (AW) if a longer arithmetic sequence is to be carried out.

e. Additional Words (AW). This state defines the period for arithmetic operations which exceed one word in duration. The duration of AW may be any number of word times up to 64. At the last word of arithmetic operations, this state always changes to "last word" state (LW).

f. Last Word (LW). The state defines the last word of the arithmetic operations which have extended for greater than one word in duration. This state changes, after one word time, to IR to cause read-in of the next instruction.

5-4. VARIATIONS OF STATES

The control states usually occur in the order given in paragraph 5-3. However, a time delay can occur between states when the sector address specified by the program does not arrive in the proper time after the control state is entered. Optimization of the program reduces this delay to the minimum when only the minimum word times for the orders are allowed in the program. The minimum word times for the various orders of operation appear in table 5-3.

Add, subtract, clear-add, and normal store are said to be first word (FW) operations. The multiply, divide, and some modified store orders are additional word (AW) operations and vary in length. Since the conditional transfer order is a non-arithmetic operation, the time required to perform it is mainly dependent upon access time.

5-5. INSTRUCTION WORD FORMAT

Each instruction word of 25 bits supplies: (1) order code of the current operation, (2) binary address of the current operand to be used, and (3) binary address of the succeeding instruction word. Table 5-1 shows the distribution of bits in an instruction word.

5-6. DATA WORD FORMAT

Each data word, also of 25 bits, is a decimal number which has been converted to binary code. The least significant digit appears at position P_{24}; the most significant digit at position P_1. P_0 represents the sign of the number, and a binary point exists between P_0 and P_1, supplying a fixed reference. See table 5-2.

5-7. ORDER FORMAT

The basic orders used in the programming of the computer appear in table 5-3. Descriptions of the circuitry involved and the sections of the computer referred to appear in chapter 4.

Table 5-1

Instruction Word Format

Tα					Tβ					Sβ						Sα						0		
P24	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
2^0	2^1	2^2	2^3	2^4	2^0	2^1	2^2	2^3	2^4	2^0	2^1	2^2	2^3	2^4	2^5	2^0	2^1	2^2	2^3	2^4	p^5	2^0	2^1	2^2

Tα = Track address of the next instruction

Tβ = Track address of the operand

Sβ = Sector address of the operand

Sα = Sector address of the next instruction

0α = Order to be performed

Table 5-2

Data Word Format

P24	P23	P22	P21	P20	P19	P18	P17	P16	P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
2^{-24}																							2^{-1}	Sign

P24 = Least significant bit position

P1 = Most significant bit position

P0 = Sign bit position

Table 5-3

Order Coding

Order	Order Code	Order Abbrev.	Order Description	Minimum Word Time
Clear Add	000	At	Clear the accumulator C (A) to 0 and to it add the contents of the beta address C (M).	4
Extract	001	Ex	Compare the contents of C (A) to contents of C (M). Resultant contains 1 where 1 occurred in both C (A) and C (M).	4
Multiply	010	Mu	Multiply the contents of C (M) by the contents of C (A)	2+n*
Conditional Transfer	011	T. C.	If the sign of C (A) is negative take the next instruction from the alpha address of the current instruction. If the sign of C (A) is positive, take the next instruction from the beta address of the current instruction.	2
Add	100	Ad	Add the contents of the beta address to the contents of the accumulator.	4
Divide	101	Dv	Divide C (A) by C (M).	3+n*
Subtract	110	Su	Subtract the contents of C (A) from the contents of C (M).	4
Store	111	Sr	Store Order Coding (see table 5-5).	
Shop		Sr	Discrete Order Coding (see table 5-6).	

* = Length of operation

5-8 COMPUTATION TIME

The times shown in table 5-4 represent minimum computation time only. Any additional access time must also be considered in estimating the time of a program.

Table 5-4

Computation and Word Times

Event	Minimum Word Times Required	Real Time Seconds
Clear Add	4	.000625
Extract	4	.000625
Multiply (25 bits)	27	.00421875
Conditional Transfer	2	.0003125
Add	4	.000625
Divide (25 bits)	28	.004375
Subtract	4	.000625
Store (Normal)	4	.000625
Store Yo	4	.000625
Clear Add Y_{01}	4	.000625
Clear Add Y_{02}	4	.000625
Clear Add M_n	4	.000625
Extract A_0M_n; Left Shift (1 place)	5*	.00078125
Left Shift (1 place)	5*	.00078125
Extract A_0M_n; Right Shift (1 place)	5*	.00078125
Right Shift (1 place)	5*	.00078125
Relay Set (Reset)	55-64	.008125-0.01

* Each additional shift requires one additional word time.

5-9. STORE ORDERS

5-10. Normal Store

The normal store orders define the codes necessary to provide placement of the accumulator data into the temporary storage tracks. A description of the coding appears in table 5-5.

5-11. Modified Store

The modified store orders define the codes for access to the sigmator, the extract and shift, and the shift only functions. Table 5-5 shows the coding for each function.

5-12. Discrete Orders

Most discrete orders, relay sets, and relay resets, are special orders which change with each application. Therefore, descriptions of these orders—other than the stop order—are not included in this section. Table 5-6 describes the discrete order coding which applies to this section.

5-13. DATA

5-14. Binary Numbers

The Binary number system is a number system similar to the decimal system except that is constructed with 2 rather than 10 as a base. Table 5-7 shows the binary numbers and their decimal counterparts. A table of powers of 2 appears in table 5-8 for reference in making conversions.

5-15. Conversion

The binary number system is used in programming the computer. Decimal number operands are converted to the natural binary system as shown in table 5-7. The least significant bit appears at the right of each binary expression. Actual numerical quantities are expressed in paragraph 5-11.

Table 5-5

Store Order Coding

STORE ORDER FORMAT

Tβ					Order		
P19				P15	P2	P1	P0
1	0	1	0	0	1	1	1
S1	S2	S3	S4	S5	01	02	03

NORMAL STORE CODE

Tβ					Order				Minimum Word Time
S1	S2	S3	S4	S5	01	02	03		
0	0	0	0	0	1	1	1	C(A) into Y_o	4
1	0	0	0	0	1	1	1	C(A) into M_{ov}	4
0	1	0	0	0	1	1	1	C(A) into M_{ox}	4
1	1	0	0	0	1	1	1	C(A) into M_{ox} and M_{ov}	4
0	0	1	0	0	1	1	1	C(A) into M_{oz}	4
1	0	1	0	0	1	1	1	C(A) into M_{ov} and M_{oz}	4
0	1	1	0	0	1	1	1	C(A) into M_{ox} and M_{oz}	4
1	1	1	0	0	1	1	1	C(A) into M_{ox} and M_{ov} and M_{oz}	4

MODIFIED STORE CODE

S1	S2	S3	S4	S5	01	02	03		Minimum Word Time
1	0	0	1	0	1	1	1	Y_o2 into A_o	4
0	1	0	1	0	1	1	1	M_n into A_o	4
1	1	0	1	0	1	1	1	Y_o1 into A_o	4
0	0	1	1	0	1	1	1	Ex A_o with M_n; Right Shift	4+n*
1	0	1	1	0	1	1	1	Right Shift Only	4+n*
0	1	1	1	0	1	1	1	E_x A_o with M_n; Left Shift	4+n*
1	1	1	1	0	1	1	1	Left Shift Only	4+n*

*A minimum of one shift is required and takes five word times. Each additional shift adds one word time.

Table 5-6

Discrete Order Coding

Tβ						Minimum Word Time
S1	S2	S3	S4	S5		
0	0	0	0	1	Stop	55
-	-	-	-	1	Discrete	-
-	-	-	-	1		-
-	-	-	-	1		-
-	-	-	-	1		-

Table 5-7

Binary and Decimal Equivalents

Binary	Decimal	Binary	Decimal
0 0 0 0	0	1 0 0 0 0 1	33
0 0 0 1	1	1 0 0 0 1 0	34
0 0 1 0	2	1 0 0 0 1 1	35
0 0 1 1	3	1 0 0 1 0 0	36
0 1 0 0	4	1 0 0 1 0 1	37
0 1 0 1	5	1 0 0 1 1 0	38
0 1 1 0	6	1 0 0 1 1 1	39
0 1 1 1	7	1 0 1 0 0 0	40
1 0 0 0	8	1 0 1 0 0 1	41
1 0 0 1	9	1 0 1 0 1 0	42
1 0 1 0	10	1 0 1 0 1 1	43
1 0 1 1	11	1 0 1 1 0 0	44
1 1 0 0	12	1 0 1 1 0 1	45
1 1 0 1	13	1 0 1 1 1 0	46
1 1 1 0	14	1 0 1 1 1 1	47
1 1 1 1	15	1 1 0 0 0 0	48
1 0 0 0 0	16	1 1 0 0 0 1	49
1 0 0 0 1	17	1 1 0 0 1 0	50
1 0 0 1 0	18	1 1 0 0 1 1	51
1 0 0 1 1	19	1 1 0 1 0 0	52
1 0 1 0 0	20	1 1 0 1 0 1	53
1 0 1 0 1	21	1 1 0 1 1 0	54
1 0 1 1 0	22	1 1 0 1 1 1	55
1 0 1 1 1	23	1 1 1 0 0 0	56
1 1 0 0 0	24	1 1 1 0 0 1	57
1 1 0 0 1	25	1 1 1 0 1 0	58
1 1 0 1 0	26	1 1 1 0 1 1	59
1 1 0 1 1	27	1 1 1 1 0 0	60
1 1 1 0 0	28	1 1 1 1 0 1	61
1 1 1 0 1	29	1 1 1 1 1 0	62
1 1 1 1 0	30	1 1 1 1 1 1	63
1 1 1 1 1	31	1 0 0 0 0 0 0	64
1 0 0 0 0 0	32		

Table 5-8

The Powers of 2

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125

5-16. Positive and Negative numbers

All positive numbers are programmed in the natural binary form as shown in table 5-9. In the sign position, a "o" denotes a positive sign and a "1" denotes a negative sign. All negative numbers are programmed in the two's complement form. The two's complement method is used primarily because the sign bit of a number can then be ignored until the end of the operation. By the use of complements, two numbers can be added together without requiring any sign tests in the logic. An example of the two's complement form of a number as it would exist in an operand appears in table 5-10.

Table 5-9

Positive Coded Operand

P7	P6	P5	P4	P3	P2	P1	.	P0
1	0	1	0	1	0	1		0

The above operand converts to: $\frac{+85}{128}$

Table 5-10

Negative Coded Operand

P7	P6	P5	P4	P3	P2	P1	.	P0
1	1	0	1	0	1	0		1

The above operand converts to: $\frac{-85}{128}$

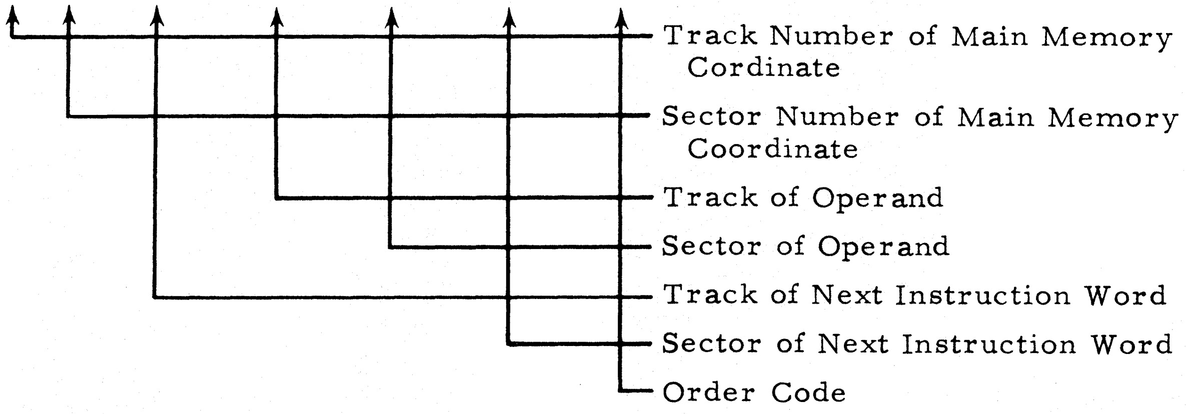
5-17. SAMPLE PROGRAM

A programmed and coded arithmetic problem appears in table 5-11 to illustrate the complete process of programming, coding into addresses, and recording on the program check sheet. The word time sequence progresses from sector to sector, with time between allowed for computation time.

Table 5-11

Programmed and Coded Problem

T	S	Tα	Tβ	Sβ	Sα	O	Comments
		Equation: $y = a_0 + a_1 \ x \ + 2_2 \ x^2$					
		Revised: $y = (a_2 \ x + a_1) x \ + a_0$					
		Assumptions: x in 28-46, 29-34; a_2 in 7-50; a_1 in 7-10; a_0 in 7-58					
		Program Addresses					
7	44	7	28	45	47	AT	Clear A_0 and add x
7	48	7	7	49	6	MU	Multiply a_2 by x (22 bits)
7	8	7	7	9	11	AD	Add a_1
7	12	7	29	33	54	MU	Multiply x by $(a_2 \, x \, a_1)$ (22 bits)
7	56	7	7	57	59	AD	Add $a_0 = y$
7	60	7	7	61	63	SR	y 28-58, 29-2, 30-30
7	0						Stop
		Coded Addresses of Program					
7	44	11100	00111	101101	111101	000	Clear - Add x
7	48	11100	11100	100011	011000	010	Multiply a_2 by x
7	8	11100	11100	100100	110100	100	Add a_1
7	12	11100	10111	100001	011011	010	Multiply x by $(a_2 \ x + a_1)$
7	56	11100	11100	100111	110111	100	Add $a_0 = y$
7	60	11100	11100	101111	111111	111	y 28-58, 29-2, 30-30
7	0						
		Note: Least significant bit is at left of each column					


Track Number of Main Memory Cordinate
Sector Number of Main Memory Coordinate
Track of Operand
Sector of Operand
Track of Next Instruction Word
Sector of Next Instruction Word
Order Code

Section VI
MAINTENANCE

6-1. GENERAL

This section contains maintenance information on the ASN-24 Computer. Computer maintenance is facilitated by use of associated test equipment which is described in the following sections of this manual. Obviously, reasonable care in the transportation, handling, and operation of the equipment will prolong its useful life and minimize the possibility of trouble. Periodic visual inspections should be made of cases, controls and switches, jacks, pins, connections, components, and interconnecting cables. Keep a log of inspections, adjustments, replacements, and repairs.

Most components are encased to protect against dirt and foreign matter; however, a vacuum cleaner may be used sparingly to clean the equipment. Never use a jet of air. The computer and associated test equipment do not require additional lubrication.

6-2. LOGIC TROUBLESHOOTING

The other sections of this manual must be studied before any attempt is made to troubleshoot the computer. A clear understanding of computer concepts, logic, and methodology is necessary for proper maintenance of the equipment. Haphazard, exploratory troubleshooting is more likely to be detrimental than corrective to complex circuitry. To determine if a particular section of the computer is in good working condition, the following logic checks should be made.

a. Check to see that the flip-flops used in a particular logical operation have complementary output signals. The two outputs of any flip-flop are complementary when one output is always the inverse of the other output; that is, if one output is high, the other output must be low. If the outputs of any flip-flop are not complementary, the flip-flop is defective and must be repaired. The converse is not necessarily true; a flip-flop may have complementary outputs and still be defective. A flip-flop must be checked dynamically to ensure that it is operating properly.

b. Dynamic check of a flip-flop requires that the logical conditions for turning it on and off must be met. (See other sections of this manual for logic.) If a flip-flop does not respond properly, it should be repaired.

c. Check the logic gates by testing the output of a logic gate for a specified pulse or DC level. If the output of a logic gate is not as specified, the inputs of the gate must be similarly checked. In checking logic networks, it must be remembered that the cross-multiplying terms qualify all flip-flop logic inputs. If the inputs are correct, the gate is faulty and must be repaired. If any inputs are not as specified, trace the incorrect input signal lines back toward their source until the defective part is located. The source of an input signal may be a manual switch, a flip-flop, or other component.

Section VII
FILL AND TEST UNIT

7-1. DESCRIPTION

The fill and test unit is 13-1/2 × 8 × 8-1/2 in., weighs 15 lbs, occupies 0.45 cu ft of space, and contains numerous switches, test points, receptacles, and associated wiring. It requires 35 watts from a 115V AC line to neutral 400-cycle three-phase source. The unit has a self-contained power supply capable of supplying all the necessary supply voltages for the fill and test unit circuitry. The unit is used, in conjunction with other equipment, to enter words or taped programs into the computer's main memory, and to test, troubleshoot, and maintain the computer.

The upper section of the front panel has a Test Start control switch which determines which of the two addresses of the instruction (β or α) is to be used as the starting address. See figure 7-1. The Stop light comes on when the fill and test control flip-flops are in the stop state, $\overline{E}_F\ (H_4 + H_5)\ \overline{E}_1\overline{E}_2$. The Step pushbutton is used to initiate all operations in either the test or fill sub-modes. The sigmator switch (B Sync 8, 16, 32, 64) is associated with the test controls on the incremental input tracks. The A Sync and B Sync switches are used for selecting particular words as the origin for the index pulse in the fill mode and origin for loop sub-mode of the test mode. Aside from this, the A Sync and B Sync switches also provide utility syncs for the oscilloscope. Numerous computer test points are on the side panels.

7-2. THEORY OF OPERATION

There are two main modes of fill and test unit operation: fill mode and test mode. The purpose of the fill mode is to enter information on the memory drum from perforated tape and to check that all entries have been correctly made. The fill mode's sub-modes are: build index (H1), tape fill (H0), check (H2), and word fill (H3). The test mode permits verification of the computer's operating condition and facilitates computer maintenance. The test mode's sub-modes are: loop (H0), one step (H1), instruction fill (H2), accumulator fill (H3), instruction sync (H4), and instruction stop (H5), A and B Sync functions are used in both the fill and test modes. See figures 7-2 through 7-5 for power supply, logic, and wiring diagrams.

7-3. A Sync (A_s)

The E_4 flip-flop is set high at P_0 of every word with q_7 (P_0 emitter follower). Flip-flop E_4 is therefore high during $S\beta$ and S_α portions of every word for coincidence detection of the A_s switch data against S_{ot}. If coincidence is detected, the E_4 flip-flop will remain high through the order portion of the word, and at q_8 (P_1 emitter follower) a sync pulse will be available at the A_s test jack on the side of the fill and test unit. This A_s pulse provides a convenient scope sync for any word around the drum; however, it is primarily used as logic control within the fill and test unit. If coincidence is not detected, the E_4 flip-flop will be set low to inhibit generation of the sync pulse at P_1.

7-4. B Sync (B_s)

B sync is generated in the same manner as A sync using E_5 as the coincidence detection flip-flop providing the sigmator switch (B Sync 8, 16, 32, 64) is in the 64 position. The sigmator switch breaks into the high order portion of the B Sync switches in order that data circulating on the sigmator tracks can be viewed on the scope. The sigmator switch enables the operator to have a B_s pulse every 8, 16, 32, or 64 words as desired. The primary function of B_s is that of a utility scope sync.

7-5. Test Mode

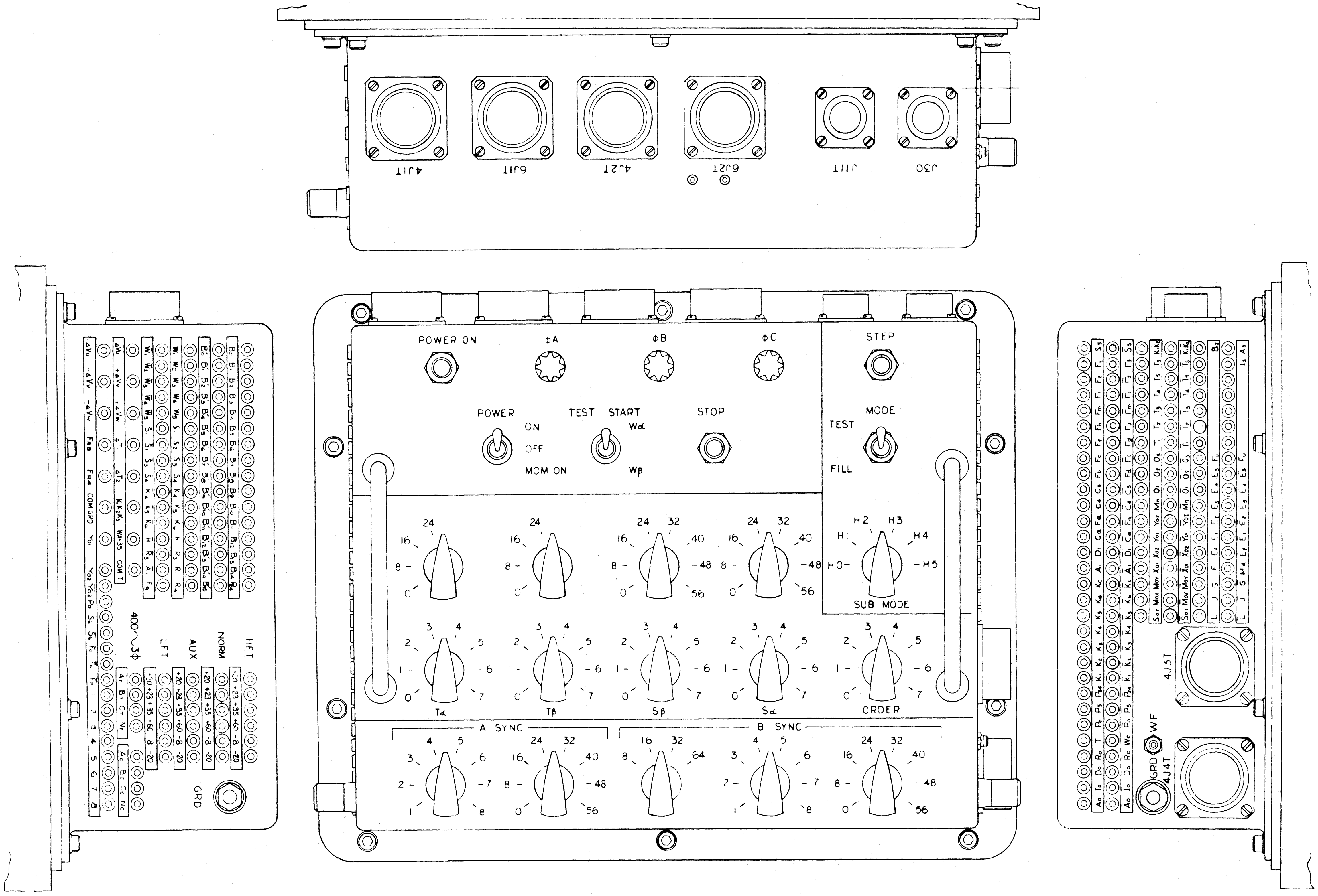


Figure 7-1. Fill-Test Control Panel

7-6. Loop (H0). Loop sub-mode is used to continuously cycle through a one-drum revolution, or less, segment of the program. This permits checking of logical operations in a dynamic manner rather than by static or single-step methods.

Upon switching into the loop sub-mode, the control flip-flops (E_1, E_2) may be in any state. However, as soon as the first A Sync (A_s) pulse occurs, the control flip-flops will be set to the $E_1\overline{E}_2$ state. The computer thus enters the stop phase and the instruction register (I_o) is filled from the instruction switches (S_1 through S_9). The computer stays in the stop phase for one word time, then goes to the Wβ or W_α phase, as determined by the Test Start switch selection. The computer follows the program from this point until the next A_s pulse occurs; then it recycles.

The sector addresses referred to above are those contained in the instruction switches. The loop length is variable up to one drum revolution. Although this sub-mode does not require the Step pushbutton to initiate cycling, depressing the pushbutton will force the computer to stay in the stop phase as long as the pushbutton is depressed. Should an internal stop occur during the loop, the computer will stay in the stop phase only until the next A_s.

Figure 7-6 indicates that due to control flip-flop E_1 (which controls instruction and order register filling) the fill operation starts at F_0, but since E_1 also goes low at the next P_0, no error is made in filling the instruction and order registers.

7-7. One Step (H1). One step sub-mode is used to cycle through the stored program one instruction per step by use of the fill and test unit Step pushbutton. Upon switching to the one step sub-mode, the control flip-flops (E_1, E_2) may be in any state. The computer is forced into the stop phase (S_p) following first instruction read-in phase (I_R) where the states of E_1 and E_2 are $\overline{E}_1\overline{E}_2$. In order to initiate one step, the Step pushbutton must be depressed and released. At the first P_0 following release of the step pushbutton, the computer enters either the

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12
2. INTERPRET ELECTRICAL & ELECTRONIC SYMBOLS PER MIL-STD-15
3. INTERPRET ELECTRICAL & ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
4. ALL DIODES 1N1909R UNLESS OTHERWISE SPECIFIED

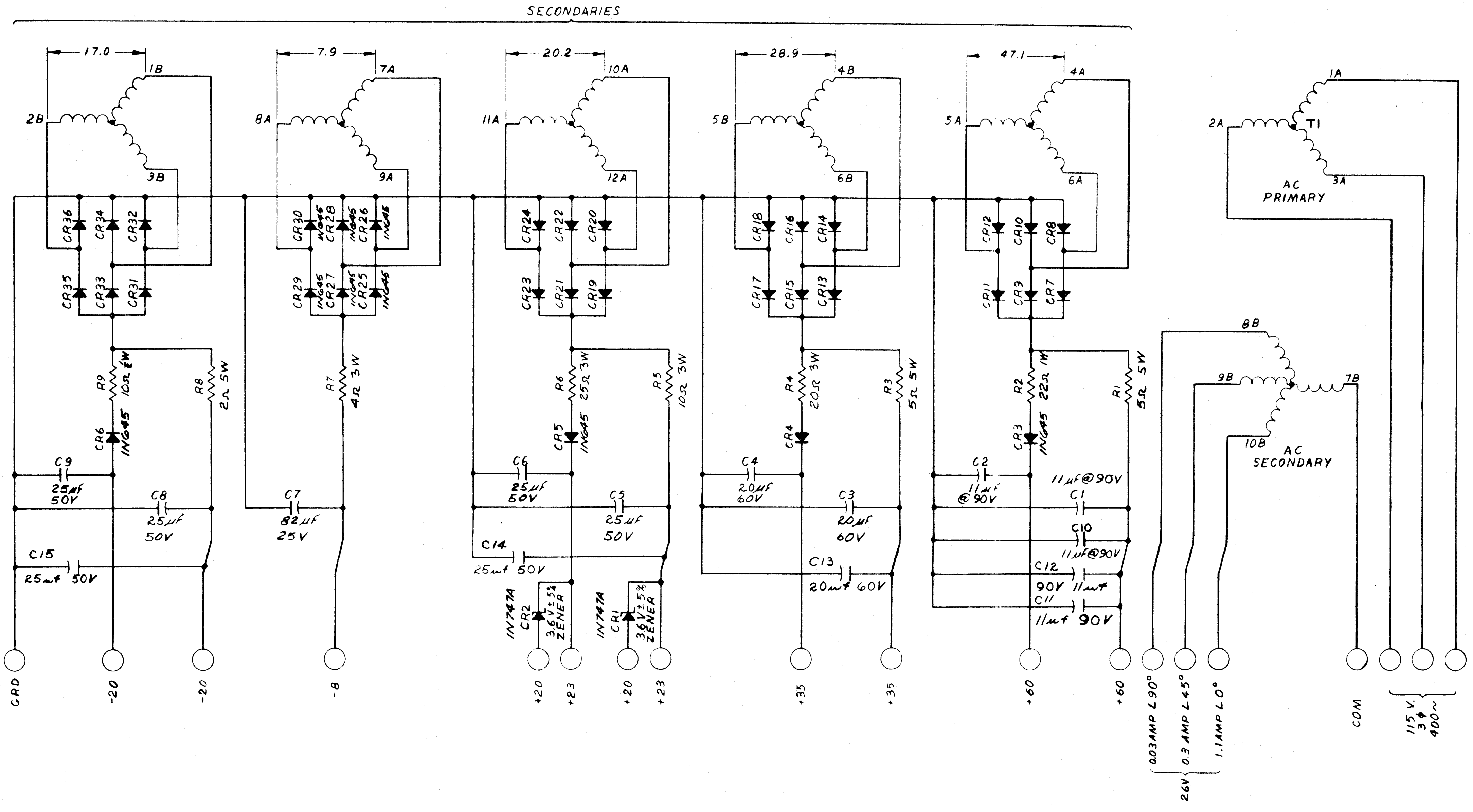
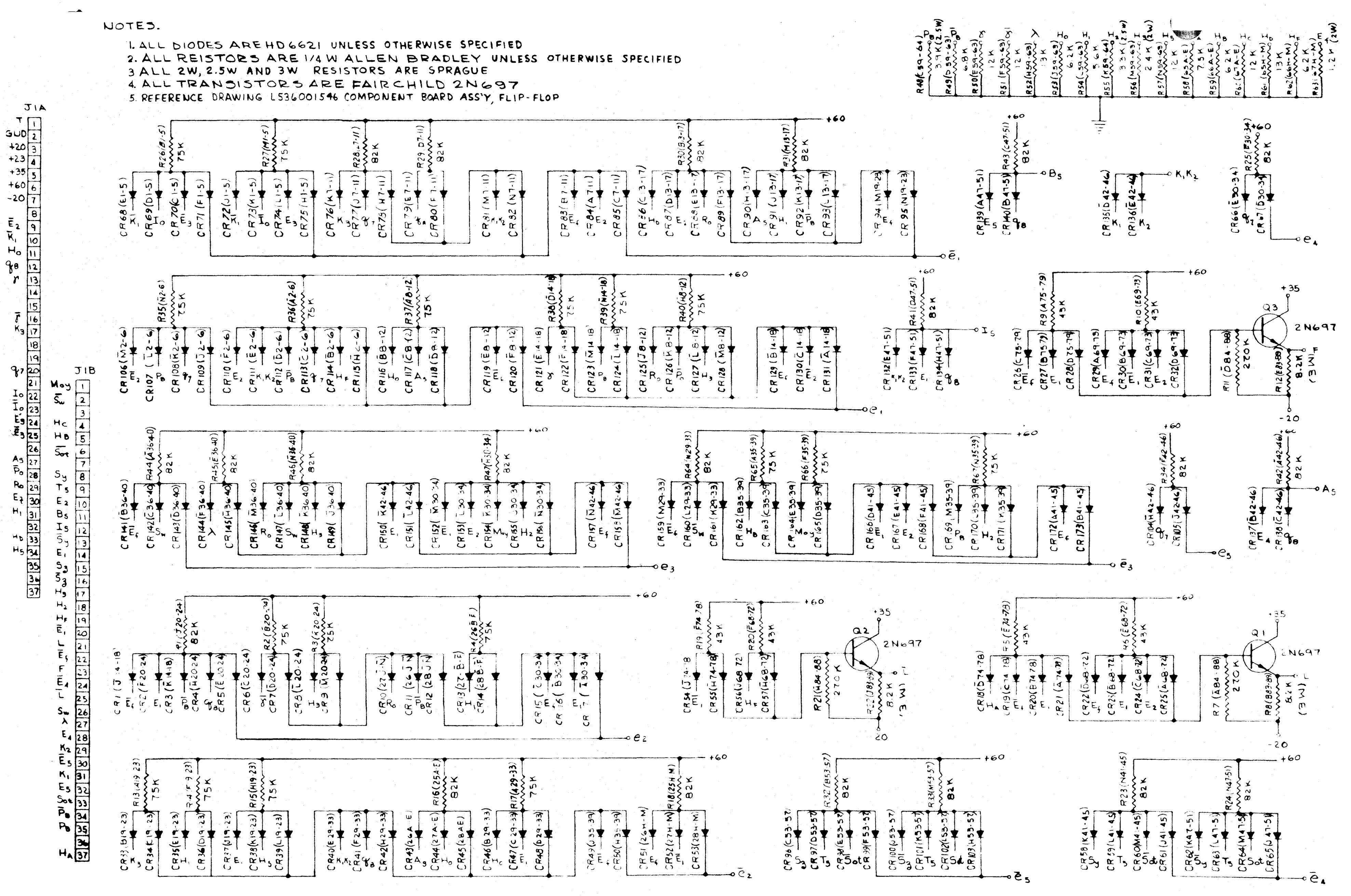


Figure 7-2. Power Supply, Schematic Diagram



NOTES.

1. ALL DIODES ARE HD6621 UNLESS OTHERWISE SPECIFIED
2. ALL REISTORS ARE 1/4 W ALLEN BRADLEY UNLESS OTHERWISE SPECIFIED
3. ALL 2W, 2.5W AND 3W RESISTORS ARE SPRAGUE
4. ALL TRANSISTORS ARE FAIRCHILD 2N697
5. REFERENCE DRAWING L536001546 COMPONENT BOARD ASS'Y, FLIP-FLOP

Figure 7-3. Logic (FF), Schematic Diagram

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12.
2. ALL DIODES ARE HD6621 UNLESS OTHERWISE SPECIFIED.
3. ALL RESISTORS ARE ALLAN BRADLEY UNLESS OTHERWISE SPECIFIED. (1/4W)
4. ALL 2W RESISTORS ARE SPRAGUE.
5. ALL TRANSISTORS ARE FAIRCHILD 2N697 UNLESS OTHERWISE SPECIFIED.
6. ALL 2N343 TRANSISTORS ARE TEXAS INSTRUMENT.
7. ASSEMBLY DRAWING REFERENCE L536 001 547.

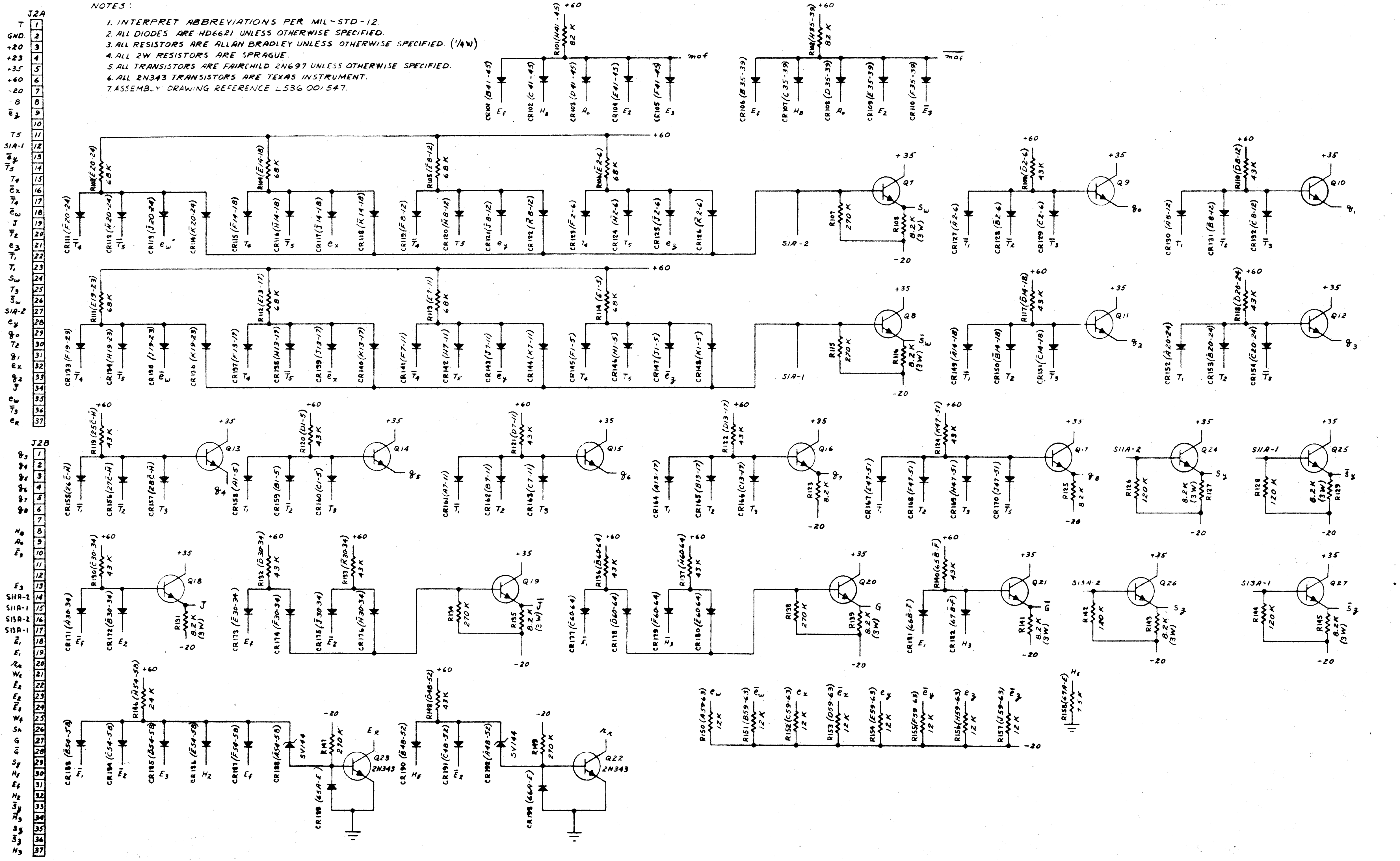
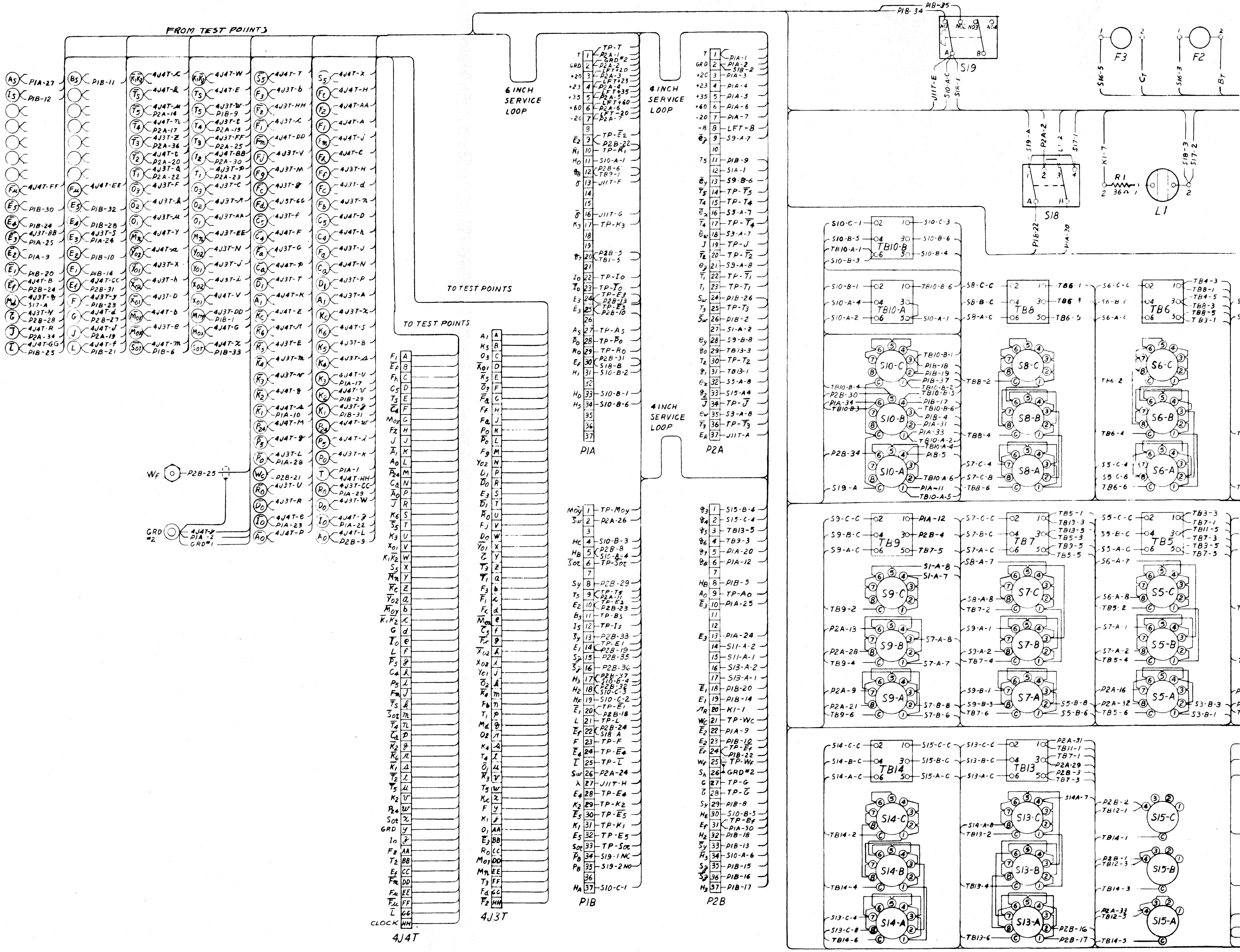


Figure 7-4. Logic (WA), Schematic Diagram

$W\beta$ or $W\alpha$ phase, as determined by the Test Start switch selection. The computer runs until the end of the next I_R phase, at which time the computer will be forced into stop (S_p) to await a step signal from the pushbutton. While the step pushbutton is held depressed, the instruction and order registers are in no way modified. At the end of a one step operation, the new instruction is contained in the instruction and order registers. Should an internal stop order (S_p) occur during a one step operation, it will be treated logically the same as an I_R phase and force the computer to stop. For control states, effective logic, and timing diagram see figure 7-7.

7-8. Instruction Fill (H2). Instruction fill provides the capability of injecting an instruction directly into the instruction register from the data word switches. All possible instructions may be selected by the data word switches.

The instruction fill sub-mode may be entered from either the one step or accumulator fill sub-modes. In either event, the control flip-flops E_1 and E_2 will be in their stop state, i. e., $\overline{E}_1\overline{E}_2$. The contents of the instruction and order registers will not be modified until the Step pushbutton is depressed. Depressing the Step pushbutton sets E_1 high and causes the instruction and order registers to be filled with the instruction defined by the instruction switches. When the Step pushbutton is released, the first P_1 pulse will set E_2 high, then P_0 will set E_1 low. Whenever both E_1 and E_2 are high at P_0, the computer will enter either the $W\beta$ or $W\alpha$ phase as determined by the Mode switch selection; although this situation may occur in this sub-mode, the computer will stay in the $W\beta$ or $W\alpha$ phase for only one word time, then be forced into the stop phase (S_p). The instruction and order registers will be filled while the step pushbutton is held depressed and through P_0 after release of the pushbutton. Instruction fill must be terminated at the end of a P_0 in order to prevent the order register shift action from terminating at a time other than P_0. For control states, effective logic, and timing diagram, see figure 7-8.



FROM TEST POINTS
TO TEST POINTS
TO TEST POINTS
6 INCH SERVICE LOOP
4 INCH SERVICE LOOP
4 INCH SERVICE LOOP
P1A
P2A
P1B
P2B
4J4T
4J3T
CLOCK
W_F
P2B-25
GRD #2
4J4T-Y
P1A-2
GRD#1
S19
S18
F3
F2
R1
36Ω
L1
TB10-B
TB10-A
TB8
TB6
TB9
TB7
TB5
TB14
TB13
S10-C
S10-B
S10-A
S8-C
S8-B
S8-A
S6-C
S6-B
S6-A
S9-C
S9-B
S9-A
S7-C
S7-B
S7-A
S5-C
S5-B
S5-A
S14-C
S14-B
S14-A
S13-C
S13-B
S13-A
S15-C
S15-B
S15-A

NOTES:
1. INTERPRET ABBREVIATIONS PER MIL-STD-12
2. INTERPRET ELECTRIAL AND ELECTRONIC SYMBOLS PER MIL-STD-15
3. INTERPRET ELECTRIAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
4. REFERENCE DRAWING: PANEL ASSEMBLY, L523 001 502

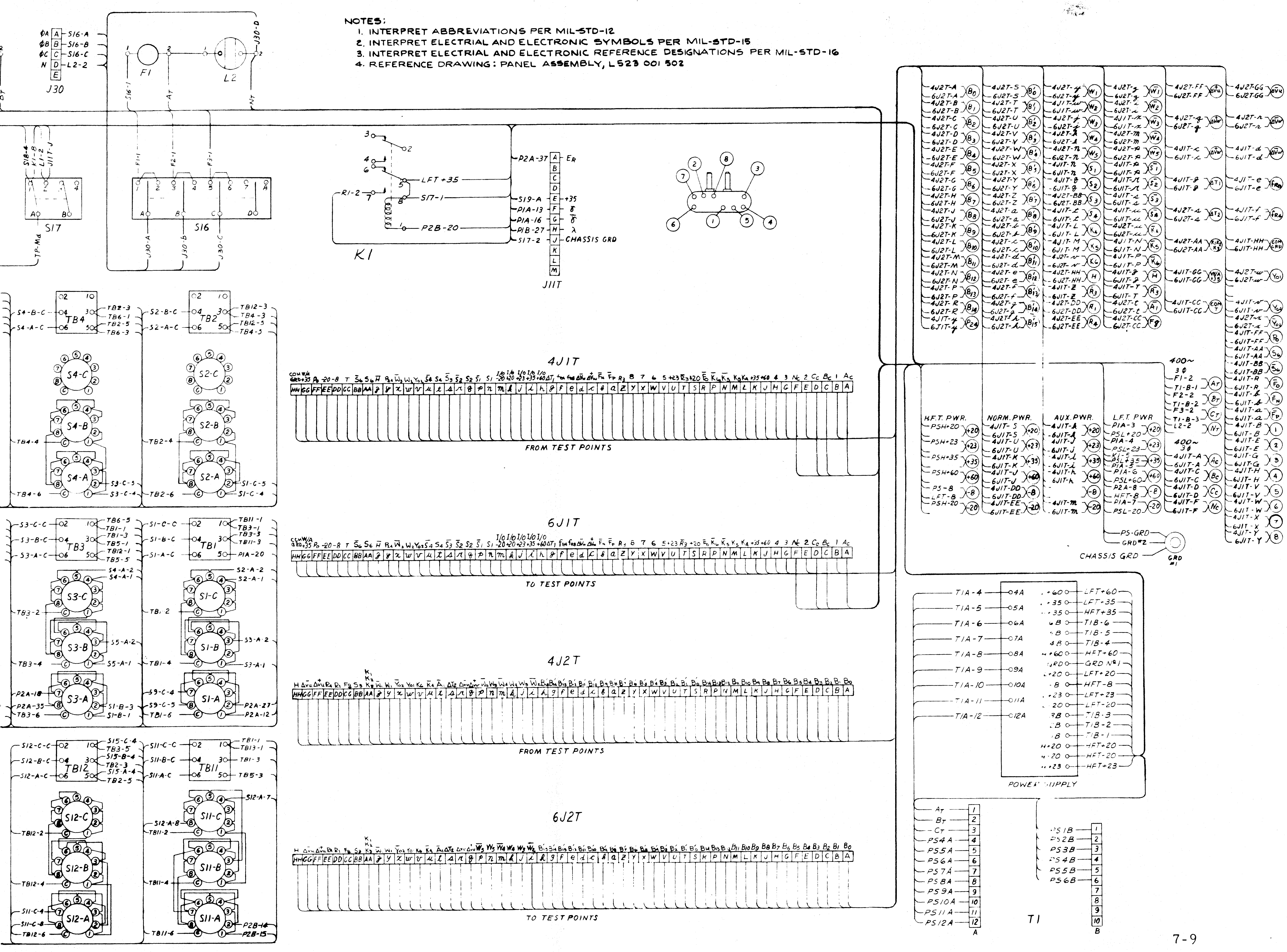
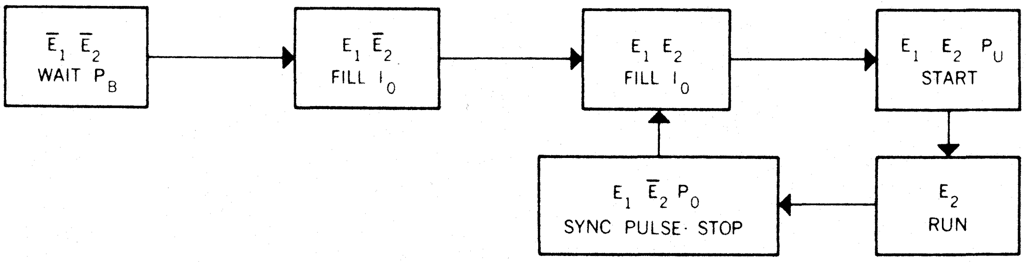


Figure 7-5.
Fill-Test, Wiring Diagram

CONTROL STATES:



EFFECTIVE LOGIC:

$e_1 = \bar{E}_2 P_B P_0 + A_S$

$\bar{e}_1 = E_2 K_1 K_2 (K_3 P_0 + P_1)$

$e_2 = E_1 \bar{P}_B P_1$

$\bar{e}_2 = A_S + K_1 K_2 K_3 P_1$

$e_S = S_W$

$\bar{e}_S = \bar{S}_W$

$f = E_1$

$j = E_2$

$l = E_1$

TIMING:

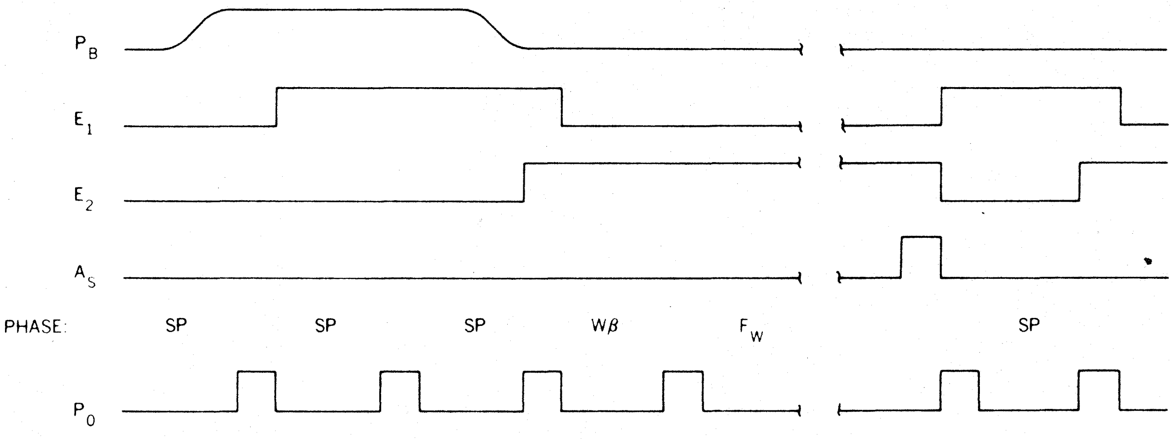
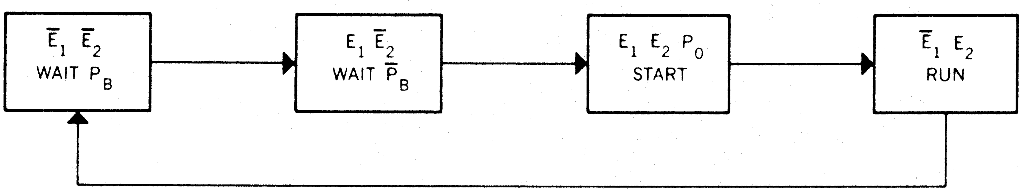


Figure 7-6. Test Mode, Loop Sub-mode

CONTROL STATES:



EFFECTIVE LOGIC:

$$e_1 = \overline{E}_2 P_B P_0 + E_2 K_1 K_2 P_0 \overline{P}_B \qquad e_S = S_W$$

$$\overline{e}_1 = E_2\left[\overline{K}_1(I_0 \overline{E}_S + \overline{I}_0 E_S) + K_1 K_2(K_3 P_0 + P_1)\right] \qquad \overline{e}_S = \overline{S}_W$$

$$e_2 = E_1 \overline{P}_B P_1 \qquad f = E_1$$

$$\overline{e}_2 = K_1 K_2 P_1 \qquad \jmath = E_2$$

TIMING:

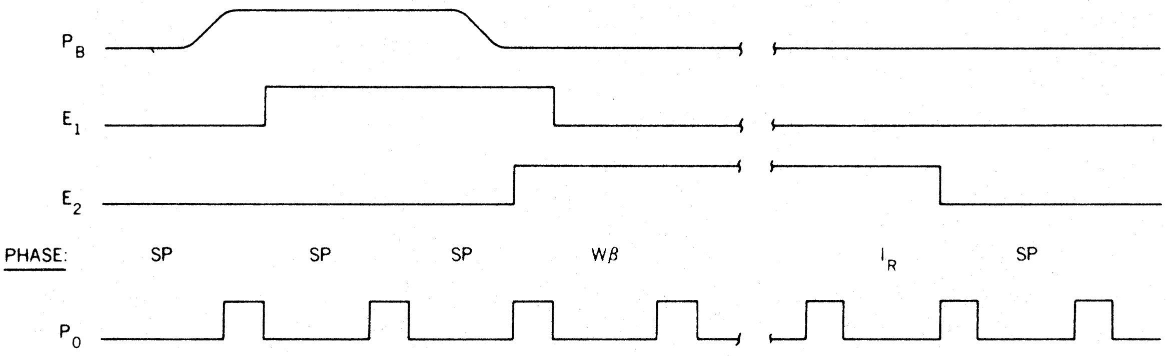
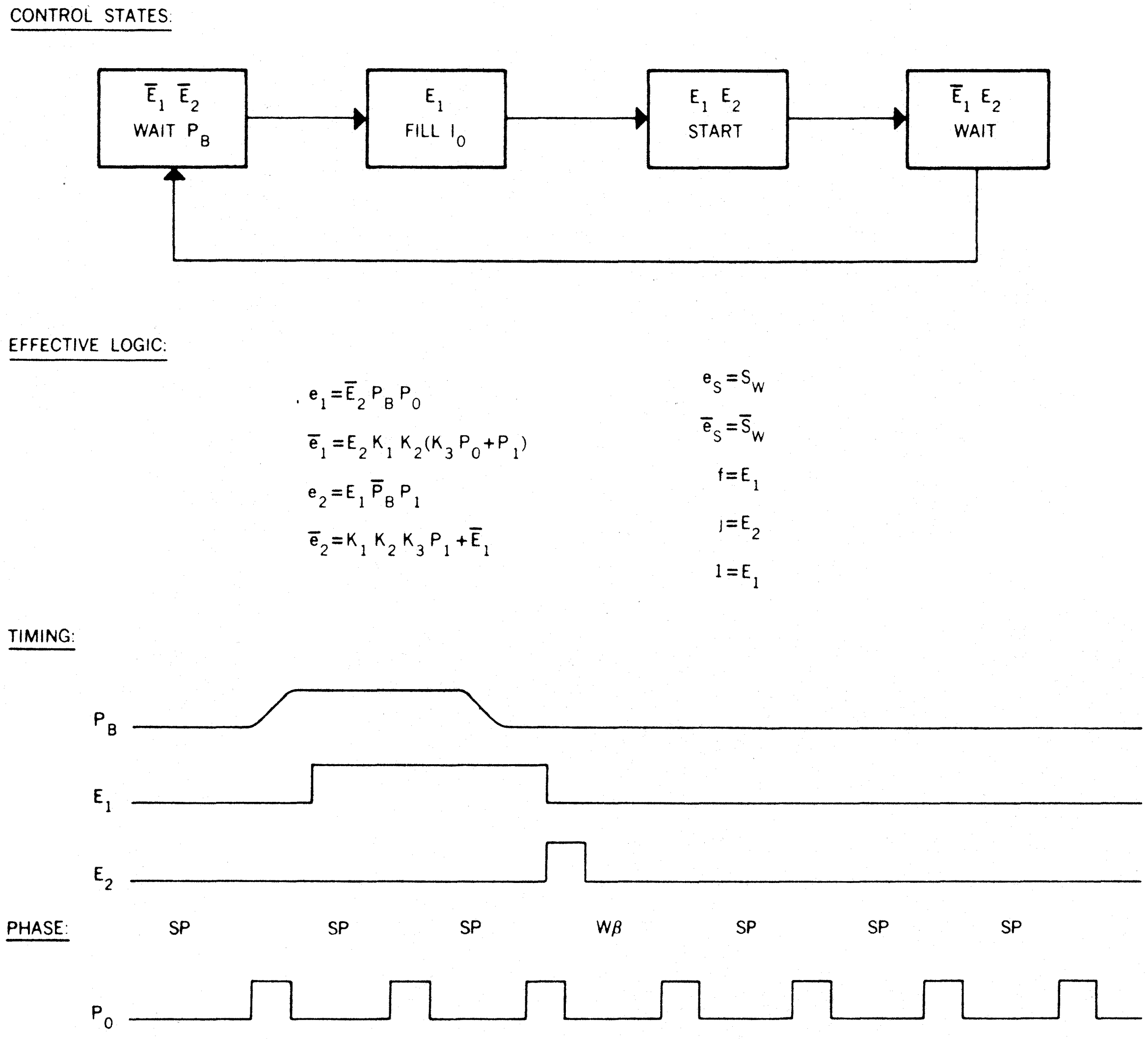


Figure 7-7. Test Mode, One Step Sub-mode



$e_1 = \bar{E}_2 P_B P_0$

$\bar{e}_1 = E_2 K_1 K_2 (K_3 P_0 + P_1)$

$e_2 = E_1 \bar{P}_B P_1$

$\bar{e}_2 = K_1 K_2 K_3 P_1 + \bar{E}_1$

$e_S = S_W$

$\bar{e}_S = \bar{S}_W$

$f = E_1$

$j = E_2$

$1 = E_1$

Figure 7-8. Test Mode, Instruction Fill Sub-mode

7-9. Accumulator Fill (H3). Accumulator fill provides the capability of injecting a data word directly into the accumulator. By accumulator and instruction fill sub-modes, it is possible to fill the accumulator from any memory location.

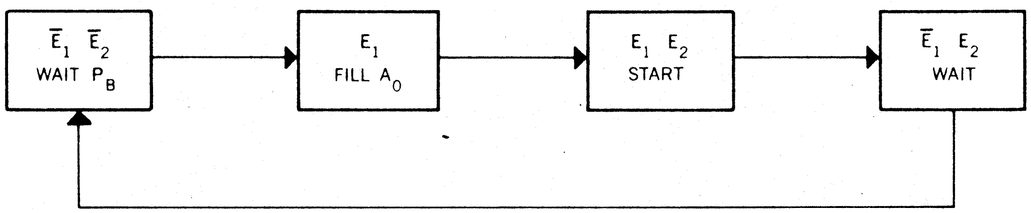
Accumulator fill sub-mode may be entered from either the instruction fill or instruction sync sub-modes. The control flip-flops (E_1, E_2) may not be in their stop states; however, E_1 and E_2 terms set F and J low after two word times to force the computer into the stop phase. The timing sequence for the control flip-flops is similar to instruction fill. The prime difference between the two sub-modes is that E_1 controls the accumulator fill logic G rather than instruction fill logic L. For control states, effective logic, and timing diagram see figure 7-9.

7-10. Instruction Sync (H4). The computer need not be in the stop phase when switching into the H4 sub-mode; however, it is likely that it will be. Assuming that E_1 and E_2 are low, depression of the Step pushbutton will cause E_1 to be set high at P_0; the first P_1 after the pushbutton is released will set E_2 high. Assuming E_1 and E_2 are high at P_0 will cause the computer to enter either the $W\beta$ or $W\alpha$ phase, determined by the Mode switch selection. Once computation has been initiated by the Step pushbutton, the computer will continue to run until an internal stop occurs or the position of the sub-mode switch is changed. Upon releasing the Step pushbutton, the program will be entered at a point defined by the instruction register contents. The instruction sync pulse I_s occurs at P_1 time. For control states, effective logic, and timing diagram, see figure 7-10.

7-11. Instruction Stop (H5). Instruction stop is used mainly in the checkout of a stored program. With this sub-mode it is possible to traverse any portion of the program and then automatically stop after executing a specified instruction.

Instruction stop sub-mode is similar to the instruction sync sub-mode with the exception that the logic term $K_1K_2E_1q_1$ which provided the sync pulse in the instruction sync sub-mode is now used to force the computer into stop phase by setting E_2 low at P_1. The shortest length of

CONTROL STATES:



EFFECTIVE LOGIC:

$$e_1 = \bar{E}_2 P_B P_0 + E_2 K_1 K_2 \bar{P}_B P_0$$

$$\bar{e}_1 = E_2 K_1 K_2 (K_3 P_0 + P_1)$$

$$e_2 = E_1 \bar{P}_B P_1$$

$$\bar{e}_2 = K_1 K_2 K_3 P_1 + \bar{E}_1$$

$$e_S = S_W$$

$$\bar{e}_S = \bar{S}_W$$

$$f = E_1$$

$$j = E_2$$

$$\bar{g} = E_1$$

TIMING:

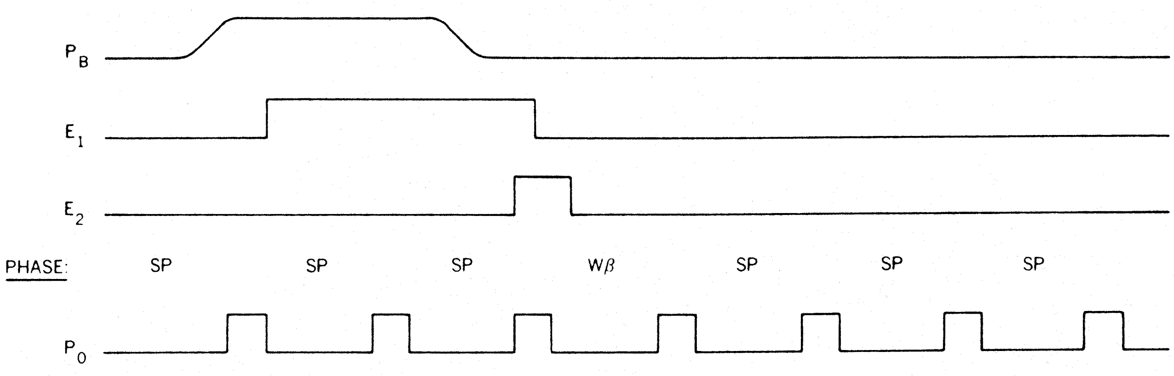
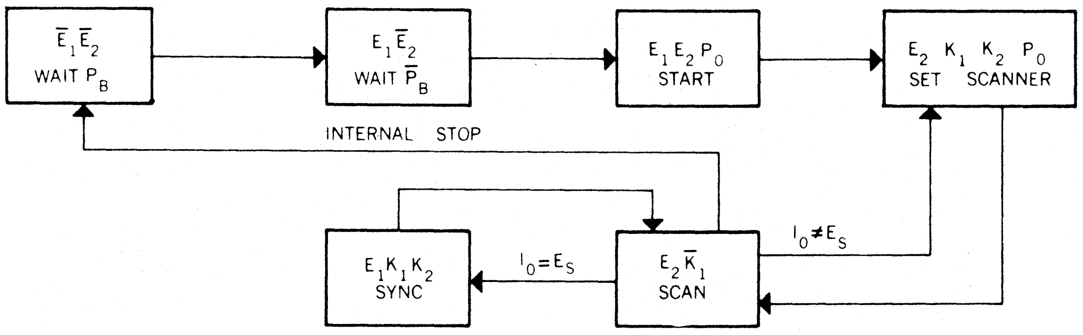


Figure 7-9. Test Mode, Accumulator Fill Sub-mode

CONTROL STATES:



EFFECTIVE LOGIC:

$e_1 = \bar{E}_2 P_B P_0 + E_2 K_1 K_2 P_0 \bar{P}_B$

$\bar{e}_1 = E_2 \bar{K}_1 (E_S \bar{I}_0 + \bar{E}_S I_0) + E_2 K_1 K_2 (K_3 P_0 + P_1)$

$e_2 = E_1 \bar{P}_B P_1$

$\bar{e}_2 = K_1 K_2 K_3 P_1$

$e_S = S_W$

$\bar{e}_S = \bar{S}_W$

$f = E_1$

$j = E_2$

$I_S = K_1 K_2 E_1 Q_1$

TIMING:

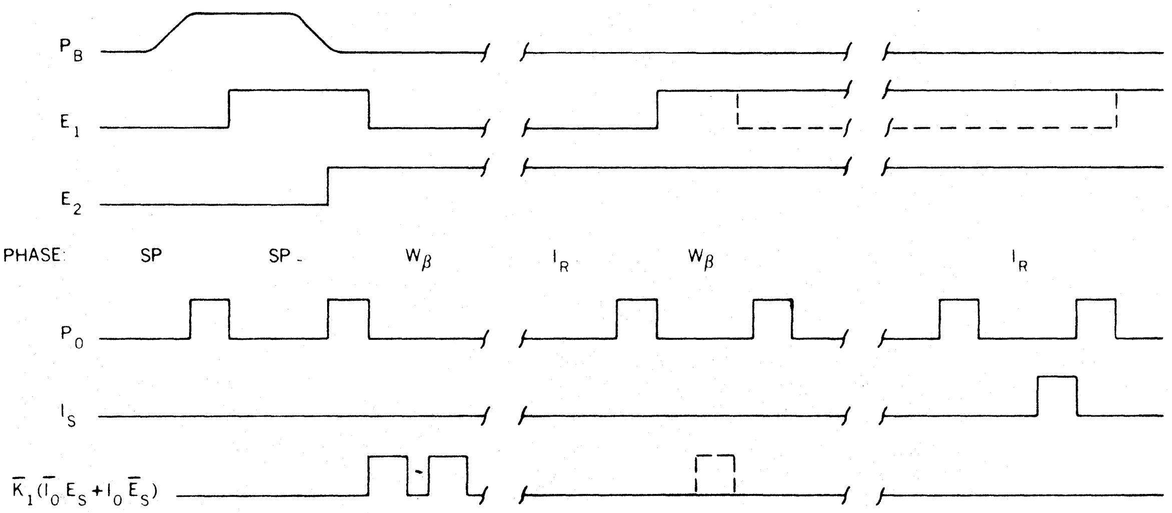


Figure 7-10. Test Mode, Instruction Sync Sub-mode

program which can be traversed in this sub-mode is two complete operations, unless an internal stop order occurs. The reason for this limitation (not important, in that a one step sub-mode is available) is that the coincidence detector is not set high until the first I_R phase after initiating this sub-mode operation. If the read-in instruction matches the instruction switches, the computer cannot stop until the following I_R phase.

In the instruction stop sub-mode, the last instruction which will be executed is the one defined by the instruction switches. For control states, effective logic, and timing diagram, see figure 7-11.

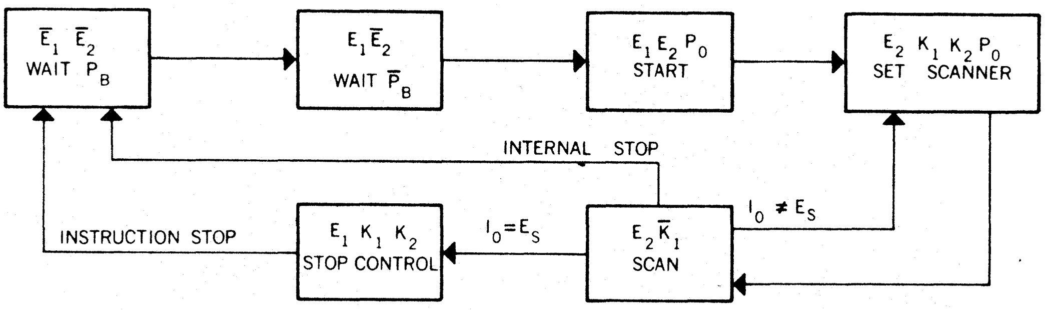
7-12. Fill Mode

7-13. Build Index (H1). In the build index sub-mode, a single indexing (or sync) pulse is written on a 1600-bit (1-drum revolution) recirculation register. The 1600-bit recirculation register is comprised of two multi-head recirculation tracks and associated read-write circuitry of the computer memory. The logic connections are accomplished by the fill and test unit.

Depressing the Step pushbutton will send E_1 and E_2 high and clear both recirculation tracks (A_0, I_0) and (D_0, R_0) to all "0's". Upon releasing the Step pushbutton, E_1 will be set low at the first A_s pulse, and E_1 being low will set E_2 low at the following pulse time. The state $\overline{E}_1E_2$ therefore occurs for only one pulse time and is used to gate the input logic to the a_0 write amplifier. This pulse will recirculate and appear at the output of the R_0 read flip-flop at P_0 and at the output of A_0 read flip-flop at P_1 every drum revolution. For control states, effective logic, and timing diagram, see figure 7-12.

7-14. Tape Fill (H0). The tape fill sub-mode receives data from the tape reader, stores it in a flip-flop, and at write time (index pulse position) supplies this single bit of tape data to a write amplifier for storage in a main memory track. Each time a bit of tape data has been stored in the main memory, the index pulse is precessed to the next bit location. This cycle is repeated 1600 times, after which the drum

CONTROL STATES:



$\bar{E}_1\ \bar{E}_2$ WAIT P_B
$E_1 \bar{E}_2$ WAIT $\bar{P}_B$
$E_1 E_2 P_0$ START
$E_2\ K_1\ K_2 P_0$ SET SCANNER
INTERNAL STOP
$I_0 \neq E_S$
INSTRUCTION STOP
$E_1\ K_1\ K_2$ STOP CONTROL
$I_0 = E_S$
$E_2 \bar{K}_1$ SCAN

EFFECTIVE LOGIC:

$$e_1 = \bar{E}_2 P_B P_0 + E_2 K_1 K_2 P_0 \bar{P}_B \qquad e_S = S_W$$

$$\bar{e}_1 = E_2 \bar{K}_1 (\bar{I}_0 E_S + I_0 \bar{E}_S) + E_2 K_1 K_2 (K_3 P_0 + P_1) \qquad \bar{e}_S = \bar{S}_W$$

$$e_2 = E_1 \bar{P}_B P_1 \qquad f = E_1$$

$$\bar{e}_2 = K_1 K_2 K_3 P_1 + K_1 K_2 E_1 P_1 \qquad j = E_2$$

TIMING:

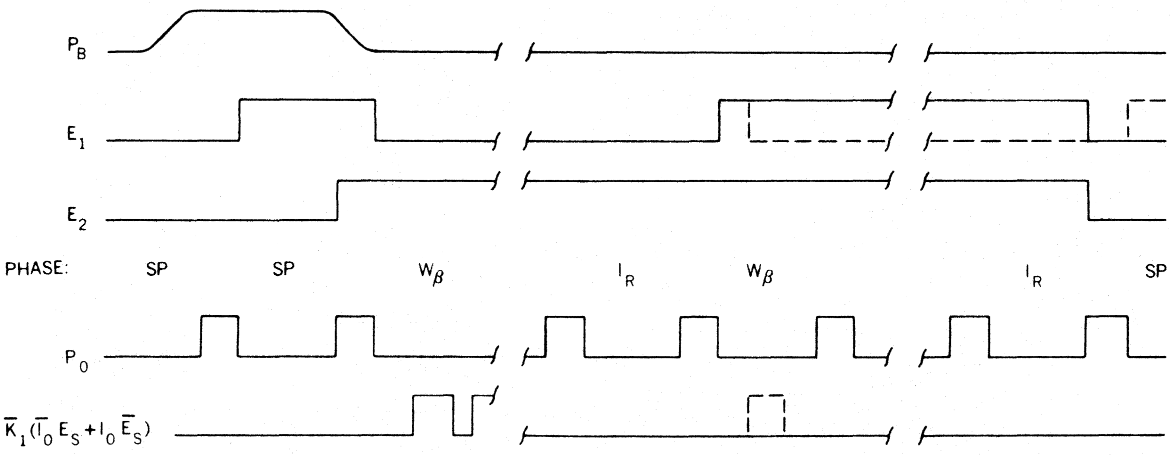
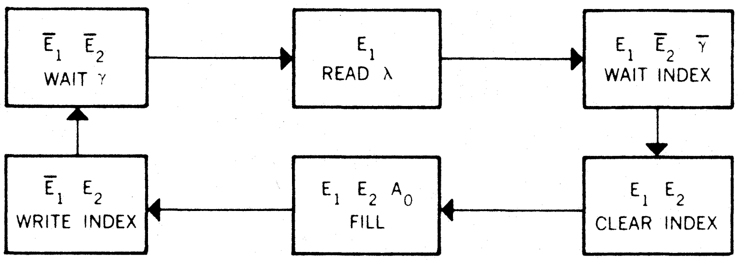


Figure 7-11. Test Mode, Instruction Stop Sub-mode

CONTROL STATES:



EFFECTIVE LOGIC:

$e_1=\bar{E}_2\ \gamma$ $\quad$ $1=E_1\ E_2$

$\bar{e}_1=E_2\ R_0$ $\quad$ $f=\bar{E}_1\ E_2$ $\quad$ $M_{of}=A_0\ E_3\ E_2$

$e_2=\bar{\gamma}\ R_0\ E_1$ $\quad$ $e_3=E_1\ \lambda$ $\quad$ $\bar{M}_{of}=A_0\ \bar{E}_3\ E_2$

$\bar{e}_2=\bar{E}_1$ $\quad$ $\bar{e}_3=\bar{E}_1\ E_2$

TIMING:

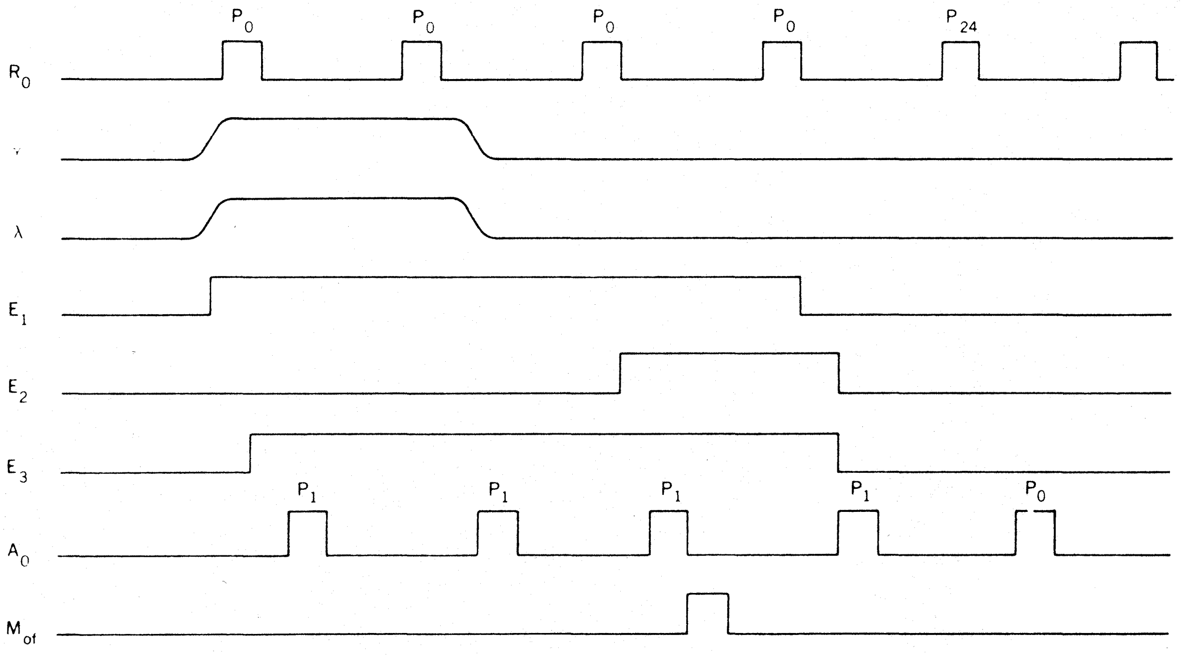


Figure 7-12. Fill Mode, Build Index Sub-mode

track is completely filled. The index pulse returns to its initial position during the final cycle.

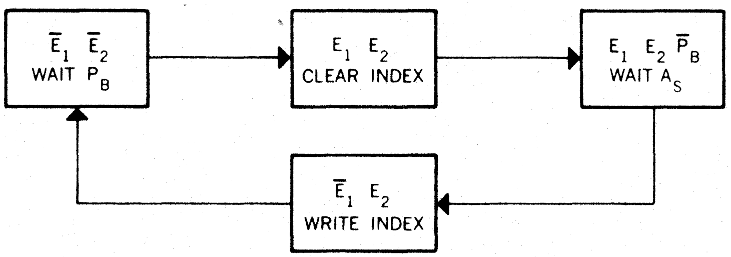
In the build index sub-mode, an index pulse is entered on the drum which is available at R_0 at P_0 time. This index digit is also available at A_0 at P_1 of the word following the R_0 index pulse. Because of a 2-bit delay (one-bit delay in the M_{of} write amplifier and one bit delay in the M_{oy} read flip-flop) it is necessary to record P_{24} tape data two bits earlier (at P_1) so that the information is available from the M_{oy} read flip-flop at P_{24} computer time. A_0 satisfies this requirement, since the index pulse is available at P_1 time to gate the tape data being held in the E_3 data flip-flop to the M_{of} write amplifier.

It is standard practice to fill the S_{ot} track first in order to accomplish pulse and sector syncronization. When the S_{ot} track has been recorded, it must be checked visually. Thereafter the A Sync switches will determine the location of the first word to be filled from either the tape fill or word fill sub-mode. For control states, effective logic, and timing diagram, see figure 7-13.

7-15. Check (H2). The check sub-mode automatically verifies the main memory contents against the corresponding tape data. Here, as in the tape-fill sub-mode, the index pulse identifies the main memory bit location. The check cycle for a single bit stores the tape data in a flip-flop and at the index pulse position compares this bit of tape data with the corresponding bit of data as read from the main memory via the main memory read amplifier, M_{oy}. If the comparison is correct, the index pulse is advanced one bit, to await the next tape bit to be verified; the cycle is then repeated. However, if the main memory data does not correspond to the tape data, the discrepancy is sensed by the error detection circuits. The error detection circuits cause the tape reader to stop and cause the error indicating light on the tape reader to light.

The index pulse is available from the output of R_0 read flip-flop at P_0. It is necessary for the check index pulse to be available at P_0 so that

CONTROL STATES:



EFFECTIVE LOGIC:

$e_1 = E_2 P_B$ $\qquad$ $\bar{e}_2 = \bar{E}_1$

$\bar{e}_1 = \bar{P}_B A_S$ $\qquad$ $1 = E_1 E_2$

$e_2 = E_1$ $\qquad$ $f = \bar{E}_1 E_2$

TIMING:

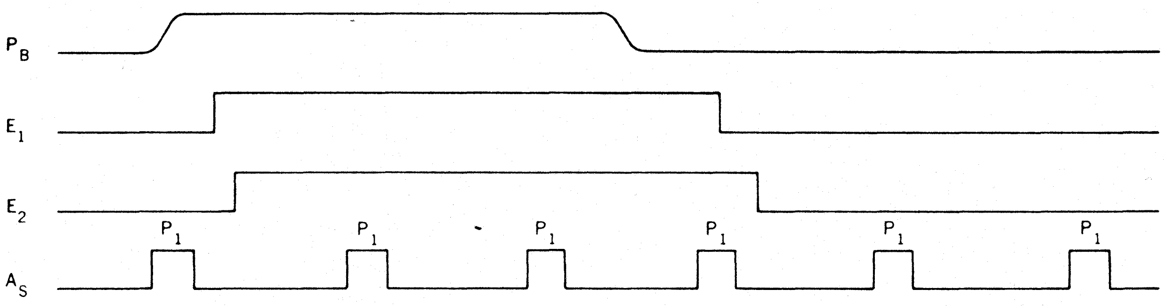


Figure 7-13. Fill Mode, Tape Fill Sub-mode

P_{24} tape data can be compared to P_{24} drum data, which has a bit delay through M_{oy} at P_{24} computer time.

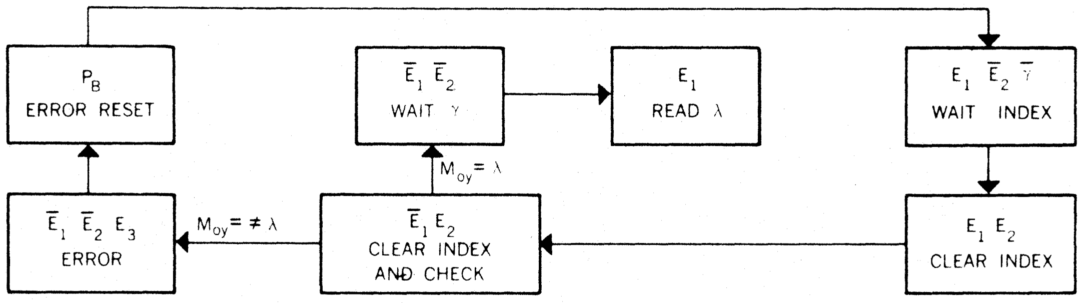
Data flip-flop E_3 is time-shared in this mode for tape data hold and error checking. The check logic on E_3 is such that if the tape data (E_3) and the stored data (M_{oy}) are equal, E_3 will be low during control state $\overline{E}_1\overline{E}_2$. If the tape data ($E_3$) and the stored data ($M_{oy}$) are not equal, E_3 will be high during control state $\overline{E}_1\overline{E}_2$. If the latter situation occurs, the error relay will be energized, the tape reader will stop, and the error light on the reader will be lit. Depressing the Step pushbutton in this sub-mode will reset the error relay and at the same time advance the index pulse one bit. For control states, effective logic, and timing diagram, see figure 7-14.

7-16. Word Fill (H3). The word fill sub-mode fills a single word in the main memory. There is no limit to the number of words which can be filled with this scheme; however, the words which are filled must be checked visually. Set-up and actual filling process is identical with tape fill, except that the data to be recorded in the main memory is taken from the instruction switches in the fill and test unit rather than from the tape reader.

The operation of word fill is similar to tape fill in that data to be recorded must be gated into the M_{of} write amplifier two bits earlier than the desired location of the data read back by M_{oy}. In this sub-mode, the data flip-flop obtains its information from the instruction switches sequentially at each R_0 pulse. Here, as in tape fill, the A_0 index pulse satisfies the aforementioned conditions and is used to gate the M_{of} write amplifier.

Initially, E_1 and E_2 are low. The A Sync switches are set to the word prior to the one to be filled and an index pulse is built. Again, as in tape fill, the sub-mode switch must be in H_1 to accomplish this. When the sub-mode switch is returned to the H_3 position, the index pulse on R_0 will occur at P_0 time, but E_1 and E_2 will remain low. To set E_1 high at P_0 either γ or P_B must be high. When the Step pushbutton is

CONTROL STATES:



EFFECTIVE LOGIC:

$e_1 = \bar{E}_2 \gamma$

$\bar{e}_1 = E_2 R_0$

$e_2 = \bar{\gamma} R_0 E_1$

$\bar{e}_2 = \bar{E}_1$

$e_3 = E_1 \lambda + \bar{E}_1 E_2 M_{oy}$

$\bar{e}_3 = \bar{E}_1 E_2 M_{oy} + P_B$

$f = \bar{E}_1 E_2$

$l = E_1 E_2$

$e_R = \bar{E}_1 E_2 E_3$

TIMING:

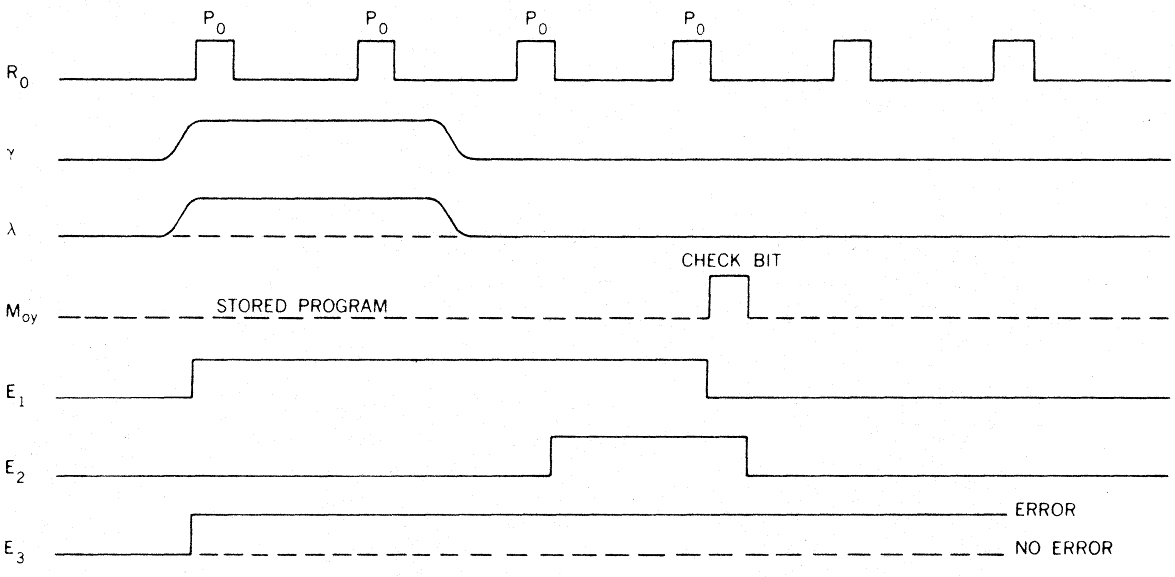
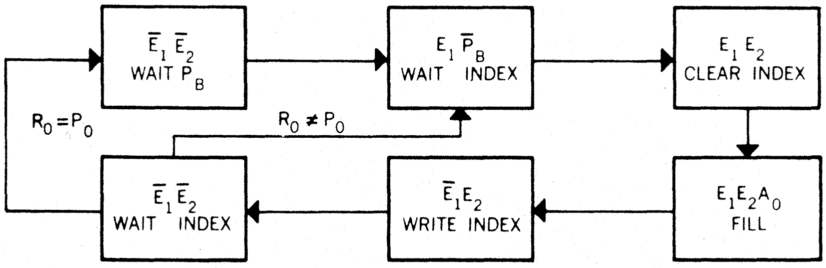


Figure 7-14. Fill Mode, Check Sub-mode

CONTROL STATES:



$\bar{E}_1 \bar{E}_2$ WAIT P_B
$E_1 \bar{P}_B$ WAIT INDEX
$E_1 E_2$ CLEAR INDEX
$R_0 = P_0$
$R_0 \neq P_0$
$\bar{E}_1 \bar{E}_2$ WAIT INDEX
$\bar{E}_1 E_2$ WRITE INDEX
$E_1 E_2 A_0$ FILL

EFFECTIVE LOGIC:

$e_1 = \bar{E}_2 P_B$ $e_3 = E_1 R_0 S_W$

$\bar{e}_1 = E_2 R_O$ $\bar{e}_3 = \bar{E}_1 E_2$

$M_{of} = A_0 E_1 E_3$

$e_2 = \bar{P}_B R_0 E_1$ $f = \bar{E}_1 E_2$

$\bar{M}_{of} = A_0 E_1 E_3$

$\bar{e}_2 = E_1$ $I = E_1 E_2$

TIMING:

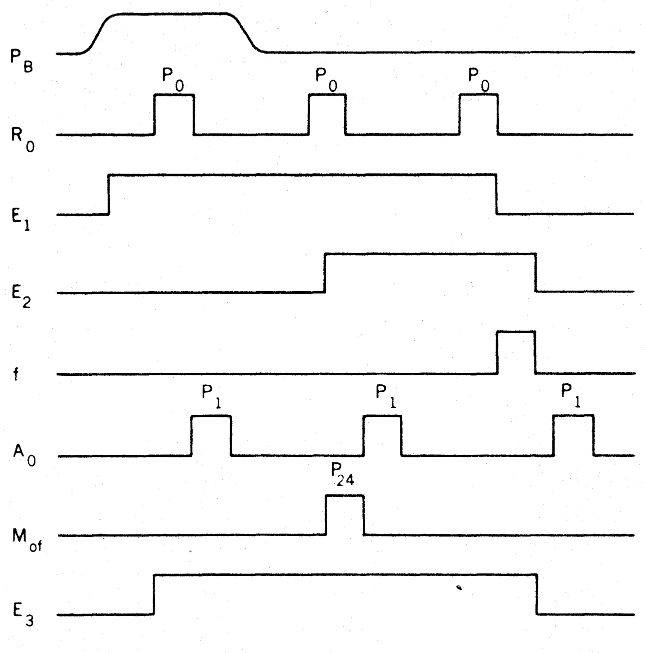


Figure 7-15. Fill Mode, Word Fill Sub-mode

depressed, the index pulse will precess one bit every two drum revolutions until the R_0 index pulse occurs at P_0. When this occurs, no new information will be recorded and the control flip-flops will remain in the $\overline{E}_1\overline{E}_2$ state. Another Step signal P_0 from the Step pushbutton is required for each additional word to be filled. For control states, effective logic, and timing diagram, see figure 7-15.

7-17. FILL AND TEST LOGIC

$$e_1 = \overline{E}_f\,(\overline{E}_2P_Bq_7 + E_2K_1K_2\overline{P}_Bq_7H_f + H_oA_s) + E_f\overline{E}_2\,(\gamma + P_B + R_o\overline{P}_oH_3)$$
$$\overline{e}_1 = \overline{E}_fE_2\,\left[\overline{K}_1I_o\overline{E}_3 + \overline{K}_1\overline{I}_oE_3 + K_1K_2\,(K_3q_7 + q_8)\right] + E_f\,(E_2R_oH_D + A_sH_1\overline{P}_B)$$

$$e_2 = \overline{E}_fE_1\overline{P}_Bq_8 + E_fE_1\,\left[(\overline{\gamma} + H_3)\,R_oP_B + H_1\right]$$
$$\overline{e}_2 = \overline{E}_f\,\left[K_1K_2q_8\,(K_3 + E_1H_5 + H_1) + A_sH_o + \overline{E}_1H_c\right] + E_f\overline{E}_1$$

$$e_3 = \overline{E}_fS_w + E_f\,\left[E_1\,(\lambda + R_o\overline{S}_wH_3) + \overline{E}_1E_2M_{oy}H_2\right]$$
$$\overline{e}_3 = \overline{E}_f\overline{S}_w + E_f\,\left[\overline{E}_1E_2\,(H_B + M_{oy}) + P_BH_2\right]$$

$$e_4 = q_7$$
$$\overline{e}_4 = S_yT_5\overline{S}_{ot} + \overline{S}_yT_5S_{ot}$$

$$e_5 = q_7$$
$$\overline{e}_5 = S_zT_5\overline{S}_{ot} + \overline{S}_zT_5S_{ot}$$

$$S_w = e_w\overline{T}_4\overline{T}_5 + e_xT_4T_5 + e_y\overline{T}_4T_5 + e_zT_4T_5 + S_oq_8 + S_{24}q_7$$
$$\overline{S}_w = \overline{e}_w\overline{T}_4\overline{T}_5 + \overline{e}_xT_4\overline{T}_5 + \overline{e}_y\overline{T}_4T_5 + \overline{e}_zT_4T_5 + \overline{S}_oq_8 + \overline{S}_{24}q_7$$

$$m_{of} = E_fH_BA_oE_2E_3$$
$$\overline{m}_{of} = E_fH_BA_oE_2\overline{E}_3$$

$$A_s = E_4q_8$$

$$B_s = E_5q_8$$

$$I_s = K_1K_2E_1q_8$$

$$e_R = \overline{E}_1\overline{E}_2E_3H_2E_f$$

$$r_R = H_E\overline{E}_2$$

$$F = \overline{E}_fE_1 + E_f\overline{E}_1E_2$$

$$L = \overline{E}_fE_1H_A + E_fE_1E_2$$
$$\overline{L} = \overline{E}_1 + H_f$$

$J = \bar{E}_f E_2$

$\bar{J} = E_f + \bar{E}_2$

$G = \bar{E}_1 + \bar{H}_3$

$\bar{G} = E_1 H_3$

7-18. LOGIC EQUATIONS DEFINING TEST AND FILL CONTROL OF COMPUTER

$k_1 = \bar{J} P_o$

$\bar{k}_1 = J P_o K_2 (F + \bar{K}_3) R_4$

$k_2 = \bar{J} P_o$

$\bar{k}_2 = J P_o \bar{K}_1 K_3 K_c (O_1 + \bar{O}_2 + \bar{O}_3)$

$k_3 = \bar{J} P_o$

$\bar{k}_3 = J P_o (K_1 \bar{K}_2 S_5 + K_1 K_2 M_d F + \bar{K}_1 K_2 K_c)$

$o_1 = L O_2$

$\bar{o}_1 = L \bar{O}_2$

$o_2 = L O_3$

$\bar{o}_2 = L \bar{O}_3$

$o_3 = L E_3$

$\bar{o}_3 = L \bar{E}_3$

$a_1 = G A_o$ (Normal Compute Logic) $+ \bar{G} E_3$

$\bar{a}_1 = G \bar{A}_o$ (Normal Compute Logic) $+ \bar{G} \bar{E}_3$

$a_o = E_f F$

$\bar{a}_o = E_f L$ Note: $\bar{E}_f$ must be high for normal operation.

$d_o = E_f I_o$

$\bar{d}_o = E_f \bar{I}_o$

$i_o = \bar{E}_f L E_3 + \bar{E}_f \bar{L} (\bar{K}_1 + K_3)$

$\bar{i}_o = \bar{E}_f L \bar{E}_3 + \bar{E}_f \bar{L} (\bar{K}_1 + K_3 \bar{R}_3) \bar{I}_o$

$x_o = E_f F$ Note: $\bar{E}_f$ must be high for normal operation.

$\bar{x}_o = E_f L$ Note: $\bar{E}_f$ must be high for normal operation.

$y_o = E_f F$

$\bar{y}_o = E_f L$

$\bar{m}_{ox} = E_f$

7-19. OPERATING PROCEDURES

The following step-by-step procedures stipulate the equipment and method used to enter a program, a word, or words in the computer and verify that the information entered on the memory drum is correct, assuming that S_{ot} has been programmed.

7-20. Equipment Necessary

a. Fill and test unit.

b. Tape reader.

c. Oscilloscope (Tektronix Type 551).

d. Two type 54C plug-in preamplifiers.

7-21. Test Preparation

a. Set Power switch on tape reader at OFF.

b. Connect power cable from JT10 to 115V AC, 400-cycle, three-phase source.

c. Connect interconnection cable L542001517 from J11 on tape reader to J11T on fill and test unit.

d. Set Power switch on fill and test unit at OFF.

e. Connect power cable L542001514 to 115V AC, 400-cycle, three-phase source

f. Connect cables L542001508, L542001510, L542001509, and L542001511 between computer and fill and test unit.

g. Start computer.

7-22. Tape Fill Procedure

a. Thread program tape through reader and adjust tape so that first hole on the eighth channel is directly above sensing fingers.

NOTE

Tape ends are labeled "Start" and "Finish".
Be sure "Start" end is threaded.

b. Remove preamplifier package located on computer housing. (For drum tracks 0 through 15, remove preamplifier package number 1. For drum tracks 16 through 31, remove preamplifier package number 2.)

c. Connect fill adapter to location in step b from which preamplifier package was removed.

d. Set the Mode switch at FILL.

e. Set Sub Mode switch at H1.

f. Set A Sync switches at sector 63 (56 plus 7).

g. Set B Sync switches at sector 63 (56 plus 7),

h. Connect oscilloscope vertical input probe to jack A_o.

i. Connect oscilloscope external sync probe to jack B_s.

j. Set Power switch on tape reader at ON.

k. Set Channel Selector switch on tape reader to desired tape channel.

l. Connect coaxial cable W_f from fill and test unit to desired drum track indicated on fill adapter.

m. Depress Step pushbutton on fill and test unit.

n. Observe that a pulse appears at P_1 time on A_0 at word 63.

o. Set Sub Mode switch at H0.

p. Depress Start pushbutton on tape reader. Tape will continue to step for approximately four minutes and will automatically stop.

q. When the tape has stopped, observe that a pulse is present at P_1 on A_0 at word 63.

r. Perform the following check procedure.

7-23. Tape Check Procedure

a. Remove coaxial cable W_f from the fill adapter.

b. Connect adapter cable to M_{oy} test jack on fill adapter.

c. Rewind tape to initial position as specified in step a of paragraph 7-22.

d. Set Sub Mode switch at H2.

e. Set Mode switch at TEST.

f. If the drum track just filled was in the 0 through 15 category, set T_β switches at 0 (0 plus 0). If the drum track just filled was in the 16 through 31 category, set T_β switches at 31 (7 plus 24).

g. Depress Step pushbutton.

h. Set Sub Mode switch at H1.

i. Set Mode switch at FILL.

j. Depress Step pushbutton.

k. Connect oscilloscope vertical input probe to jack R_o. Observe that a pulse is present at P_o time of word 63.

l. Set Sub Mode switch at H2.

m. Depress Start pushbutton on tape reader. The tape will continue to step for approximately four minutes if no error is detected. If an error is detected, the error lamp will light and the tape will halt. Repeat fill procedure if an error is detected.

n. When the tape is automatically stopped without error detection, observe that a pulse appears at P_o on R_o at word 63.

o. Repeat the fill and check procedures for each desired drum track until the program is completed.

7-24. Word Fill Procedure

NOTE

> The word fill sub-mode is used to fill any single word in the main memory and is independent of the tape reader. The words which are filled must be checked visually.

a. Prepare fill and test unit as in paragraph 7-21.

b. Set A Sync switches to one word prior to word being filled.

c. Set B Sync switches to one word prior to word being filled.

d. Depress Step pushbutton.

e. Observe that a pulse appears at P_1 time on A_0.

f. Set B Sync switches to word being stored.

g. Set word instruction switches for appropriate information.

h. Set Sub Mode switch at H3.

i. Depress Step pushbutton.

j. Observe oscilloscope. A "1" should be present at P_1 time on A_0.

k. Set Sub Mode switch at H1.

l. Repeat steps b through k for additional words to be filled.

m. Perform the following check procedure.

7-25. Word Fill Check Procedure

a. Remove coaxial cable from the fill adapter.

b. Connect adapter cable to M_{oy} jackon fill adapter.

c. Set Sub Mode switch at H2.

d. Set Mode switch at TEST.

e. Connect adapter end of cable to adapter jack of drum track to be checked.

f. If the drum track's word just filled was in the 0 through 15 category, set T_β switches at 0 (0 and 0). If the drum track's word just filled was in the 16 through 31 category, set T_β switches at 31 (7 plus 24).

g. Connect oscilloscope to M_{oy} jack on fill and test unit.

h. Set B Sync switches at word being checked.

i. Depress Step pushbutton.

j. Observe oscilloscope for correct programmed word.

h. Repeat steps a through j for additional words to be checked.

7-26. DEFINITION OF SYMBOLS

M_d = Fill andtest Wα entry control, normally low

E_1 = Time-shared control and coincidence flip-flop

E_2 = Control flip-flop

E_3 = Time-shared data hold and error detecting flip-flop

F = Fill and test Wβ or Wα entry control, normally low

G = Fill and test accumulator control, normally high

$\overline{G}$ = Fill and test accumulator control, normally low

J = Fill and test compute control, normally high

$\overline{J}$ = Fill and test stop control, normally low

L = Fill and test instruction fill control, normally low

$\overline{L}$ = Fill and test instruction fill control, normally high

E_f = Fill mode indicator, normally low

$\overline{E}_f$ = Test mode indicator, normally high

a_1 = Serial adder flip-flop, logic input

a_0 = Accumulator write amplifier, logic input

E_R = Error relay driver output

λ = The "1's" tape data logic signal

γ = The read tape signal from reader

$\overline{\gamma}$ = The wait tape signal from reader

P_B = Logic signal from Step pushbutton

S_W = The composite instruction switches logic

H_N = Logic signals from the Sub Mode switch, defining singular or a multiple of sub-modes

q_N = Switch drive signals defining the states of T_1, T_2, T_3, and T_4

m_{of}= Logic input to the fill write amplifier

W_f = Output of the fill write amplifier

A_S = A P_1 sync pulse occurring in the A Sync switch selected word

B_S = A P_1 sync pulse occurring in the B Sync switch selected word

I_S = A P_1 sync pulse occurring during the I_R phase immediately following an operation wherein the contents of the instruction register are identical to the reading of the instruction switches

E_4 = Comparator flip-flop for A_S

e_R = Logic input to the error relay driver

r_R = Logic input to the fill and test stop relay driver

E_5 = Comparator flip-flop for B_S

$H_A = H_0 + H_2$

$H_B = H_0 + H_3$

$H_C = H_2 + H_3$

$H_D = H_0 + H_2 + H_3$

$H_E = H_4 + H_5$

$H_F = \overline{H}_0 \overline{H}_2$

e_w = Data word switch bits occurring during $\overline{T}_4 \overline{T}_5$

e_x = Data word switch bits occurring during $\overline{T}_5 T_4$

e_y = Data word switch bits occurring during $T_5 \overline{T}_4$

e_z = Data word switch bits occurring during $T_5 T_4$

S_y = A_S sync switch bits

S_z = B_S sync switch bits

Section VIII
TAPE READER

8-1. DESCRIPTION

The motorized tape reader is a self-contained unit used to read information from pre-punched paper or mylar tape (see figure 8-1). The reader has the capacity for handling tapes containing as many as eight channels. The reader is used in conjunction with the fill and test unit for entering program information on the memory drum of the computer.

The dimensions of the tape reader are 20-1/2 × 13 × 11 in. and the unit occupies 1.7 cu ft of space. The weight of the tape reader is 29 lbs, which includes the 15-lb cabinet. Power requirements are 75 watts, 208V, 40 cycles, 3 phase, Y-connected.

8-2. THEORY OF OPERATION

8-3. General

The tape reader is a pin-sensing device which senses one tape character every 125 milliseconds. The sensing pins, which are under light spring tension and are designated sequentially as 1 through 8, are cam-driven into engagement with the tape while the tape is intermittently stopped. If a hole exists over a pin, the pin moves through the tape to a raised position, allowing an associated contact assembly to operate. If there is no hole over the pin, movement of the pin is blocked by the tape, and the associated contact assembly cannot operate. The tape is advanced one step each cycle by a pinwheel which is automatically stepped immediately after the reading pins have been withdrawn from the tape.

REFERENCE DRAWINGS:
A. TAPE READER ELECTRICAL SCHEMATIC DRAWING NO. L 200 003 151
B. TAPE READER INTERNAL WIRING DIAGRAM DRAWING NO. L 200 000 074.

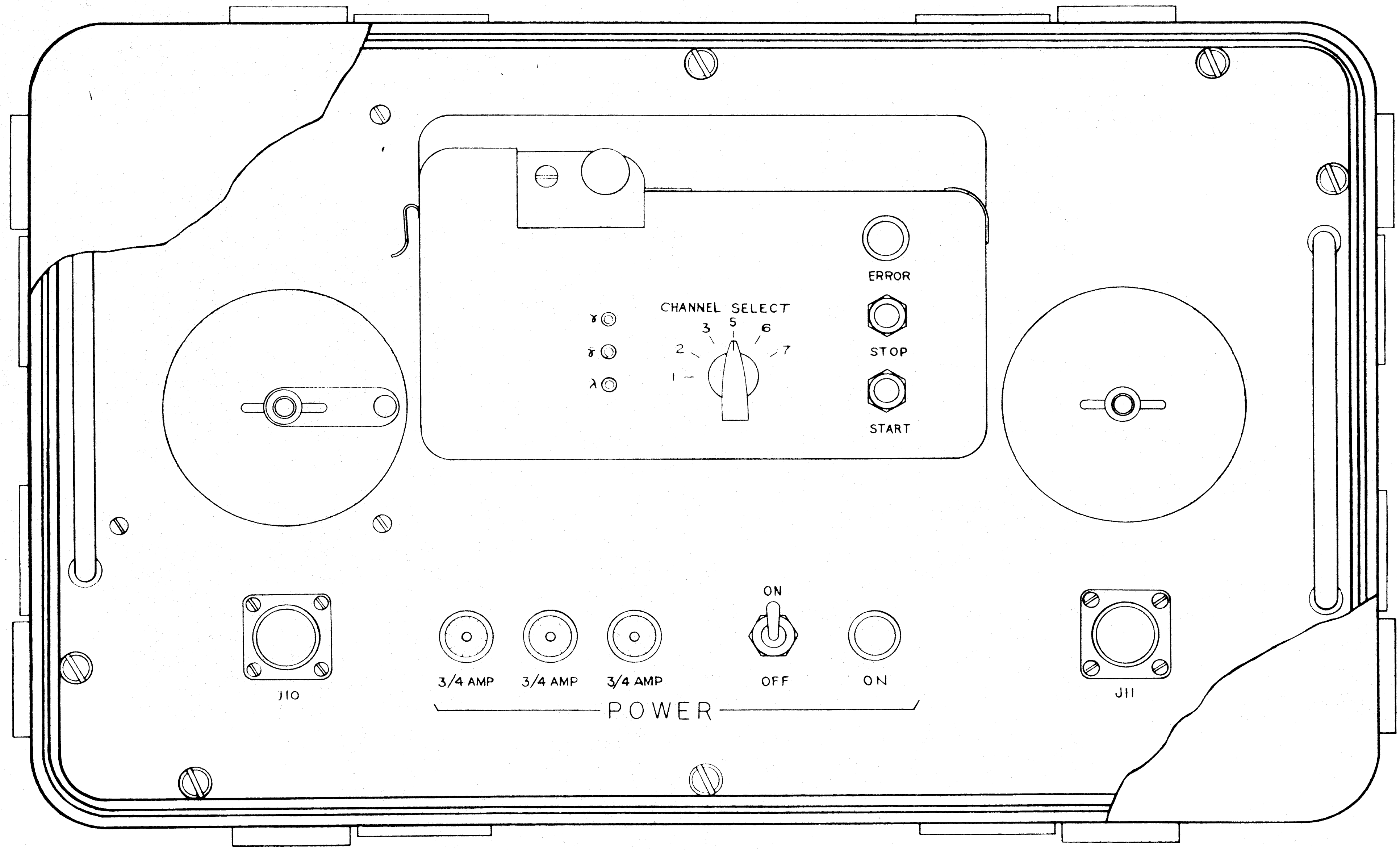


Figure 8-1. Tape Reader Control Panel

Tape channels 1, 2, 3, 5, 6, and 7 are information channels. Channels 4 and 8 are control channels. A perforation in an information channel signifies a true condition (equivalent to a "1" in the computer); absence of a perforation signifies a false condition (equivalent to a "0" in the computer). Control channels 4 or 8 must contain a perforation to permit the tape reader to continue stepping; a character lacking perforations in both channels 4 and 8 will de-energize the clutch and stop the tape.

A perforation in channel 8 also performs a second function through another set of contacts which informs the fill and test equipment that information is being sent. Transmission of information is signified by a high γ signal (perforation in channel 8); non-transmission of information is signified by a high $\bar{\gamma}$ signal (no perforation in channel 8). When γ is high, the fill and test unit accepts λ information and allows precession of the index pulse. When $\bar{\gamma}$ is high no information is sent and there is no precession of the index pulse.

Sprocket holes between channels 5 and 6 are used to drive the tape from one character to the next. Tape advance is accomplished by the engagement of a clutch to a continuously-operating motor, B1. Since there are 1600 bit positions on each track on the computer drum, tape must advance 1600 characters for each channel of data plus the number of channel 4 tape-advance holes. The time required to fill one track is approximately four minutes.

8-4. Start

When the Power On-Off switch is set in the ON position, three-phase power is applied to motor B1 and phase A power is applied to rectifier D1 and filtering circuits to produce +100V DC output, as shown in figure 8-2. Motor B1 drives a gear train and slip clutch which drives the take-up spool. When the Start pushbutton is depressed, 100V AC power is applied to coils of relay K2 and the clutch magnet via closed interlock contact 9, normally closed contacts of relay K1, normally closed contact of Stop switch S3, and normally open contacts of Start

Switch S2. Relay K2 and the clutch magnet are held in their energized states through K2 contacts 3 and 5. With the energizing of the clutch magnet, the motor shaft engages the camshaft to start a reading cycle.

8-5. Reading Cycle

Contacts 1, 2, 3, 5, 6, and 7 are controlled by sensing pins which sense correspondingly-numbered tape channels for perforations. By the position of the channel selector switch, and in accordance with the tape identification, one channel is selected to pass information to the fill and test unit. The information signal is designated as λ . When a perforation exists in the selected tape channel, its associated contact makes, during the reader cycle, to produce a high (+35V DC) λ signal which is equivalent to a "1". If no perforation exists, the associated contact remains open; thus, λ represents an open circuit condition or "0" to the computer.

Camshaft control contact 10 makes and breaks during each reading cycle. During the portion of the reading cycle in which the sensing pins are raised, contact 10 closes to apply +100V DC power via normally closed contacts 8 and 4 to the coil of relay K2 (pin 8). Since +100V DC is applied to pin 4 of relay K2, K2 will be de-energized, as will the clutch magnet, and the reader will stop. A perforation in either channel 4 or 8 will prevent this path, thus ensuring an additional reader cycle.

When γ is high, the fill-test unit accepts λ information. When $\bar{\gamma}$ is high, the fill and test unit ignores λ .

8-6. Stop

The tape reader will stop when there is no perforation of channels 4 or 8 in a particular tape character. The reader will also stop when the Stop pushbutton is depressed or when the reader receives a ground level E_R signal from the fill and test unit which will energize relay K1 to light lamp I_2 and remove +100V DC power from relay K2 and the clutch magnet.

8-7. Check

The reader, in conjunction with the fill and test unit, is equipped with a check mode. After the tape has been run, it is disengaged from the threaded locations, rewound by use of the handcrank, threaded, and run again. (If an error is detected, the error light is energized and the reader stops stepping.) The time required to check one track is approximately four minutes.

8-8. Controls

The Channel Select switch on the front of the reader is used to select the channel to be read from a tape (see figure 8-1). The Power On-Off switch is the master power switch for the tape reader. The Start pushbutton start s the reader stepping. The Stop pushbutton stops the reader. There are three front-mounted 3/4-amp fuses.

8-9. Operation

a. Set power switch at OFF.

b. Connect power cable from JT10 to a 115V AC, 400-cycle, 3 phase source.

c. Connect interconnection cable from JT11 to fill and test unit.

d. Thread tape through reader guides and secure the free end to the spool located on the right when facing the reader. Position the tape so that the first hole on the eighth channel is aligned with the sensing pins. Close Interlock switch.

NOTE

Tape ends are labeled "Start" and "Finish".
Be sure tape is threaded from "Start"end.

e. Set Channel Select switch and turn power on.

f. To stop, turn off power or press Stop pushbutton.

8-10. MAINTENANCE

Keep the reader clean; however, do not disassemble the equipment merely to clean it. Troubleshoot using figures 8-2 and 8-3.

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12.
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15.
3 INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16.
4. REFERENCE DRAWINGS:
   A. TAPE READER INTERNAL WIRING DIAGRAM L 200 003 074.
   B. BOARD COMPONENT ASSEMBLY L 536 001 527.
   C. TAPE READER ASSEMBLY L 200 003 094.
   D TAPE READER ALTERATION L200 003 083

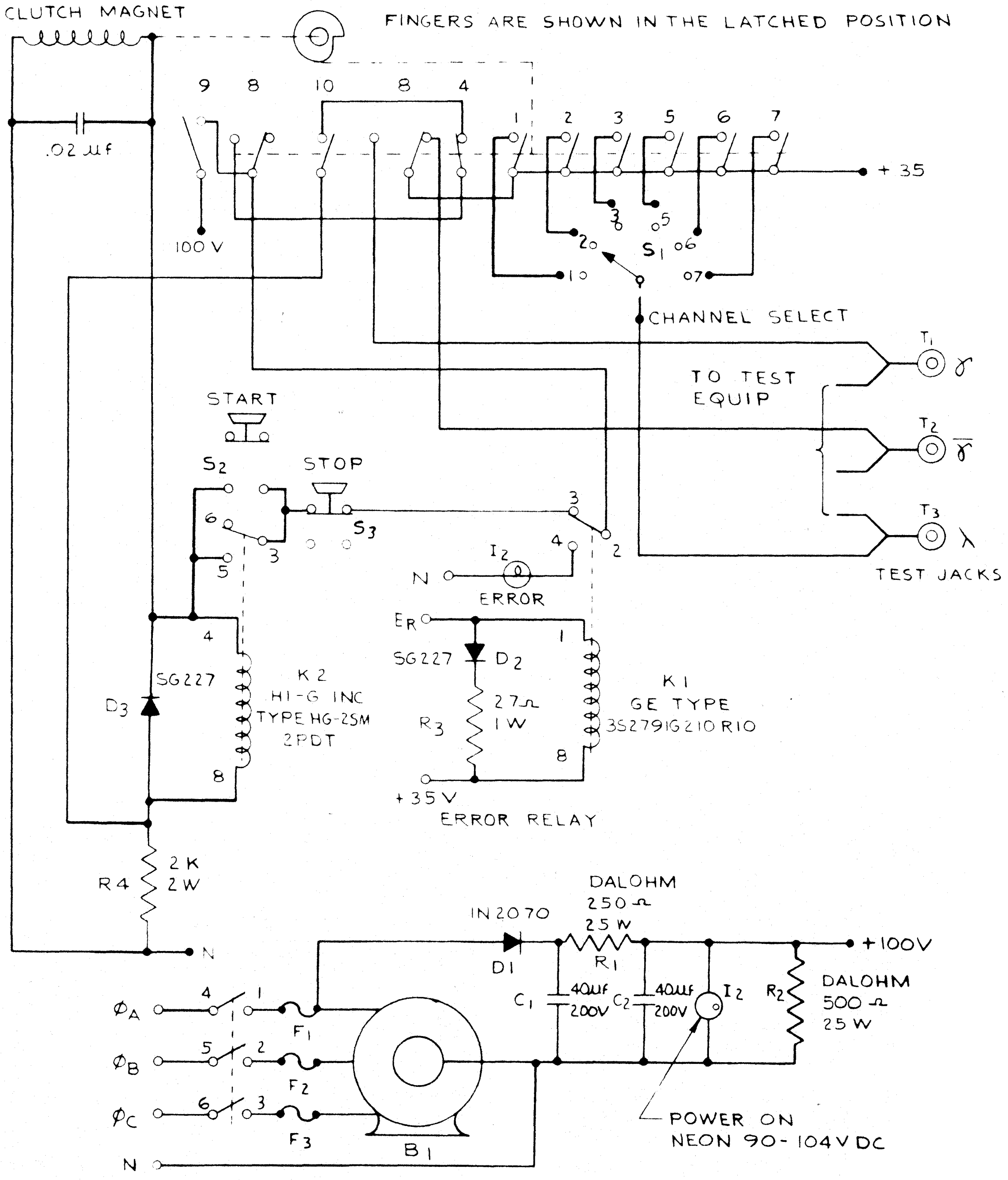
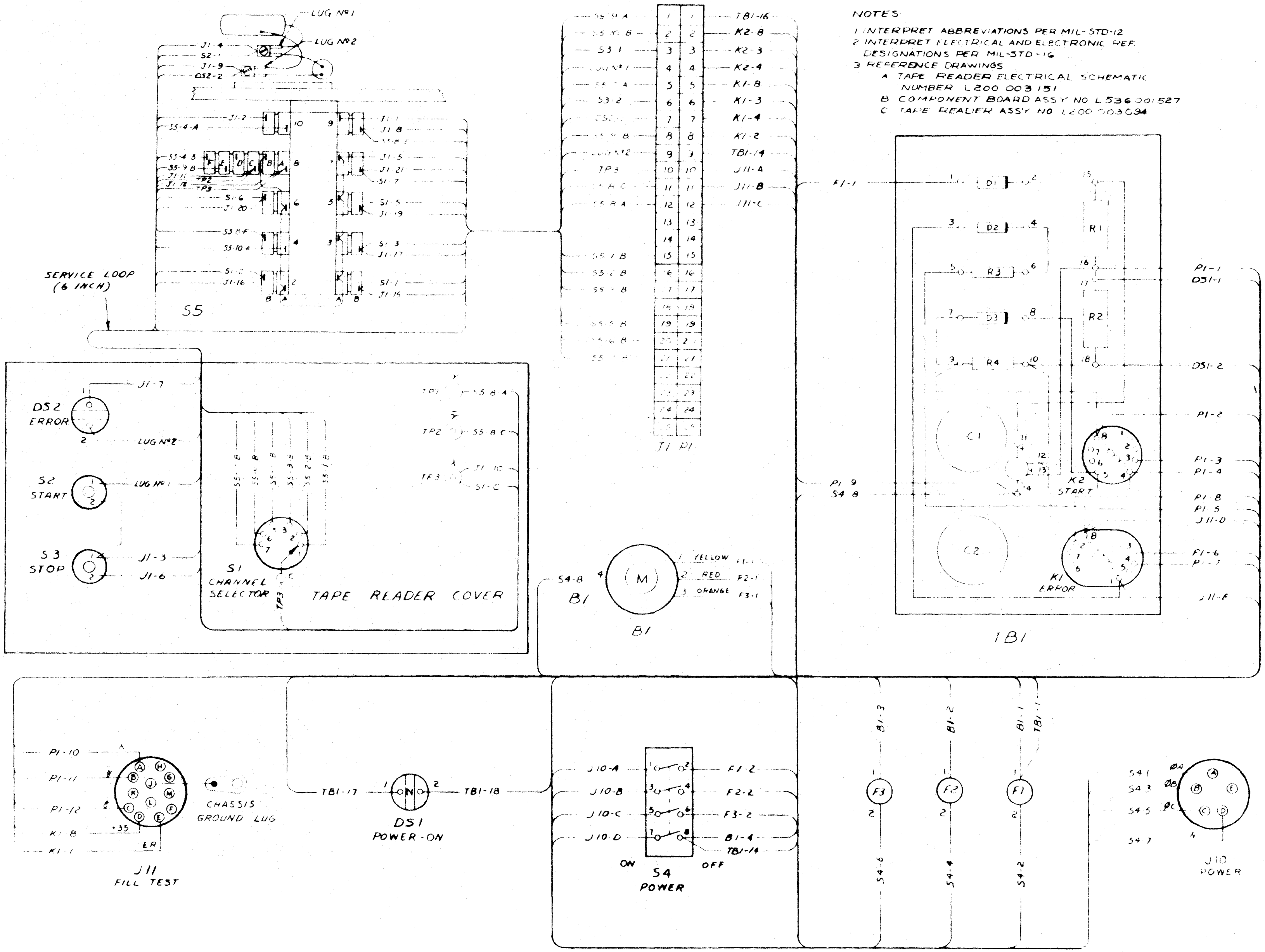


Figure 8-2. Tape Reader, Schematic Diagram



NOTES

1 INTERPRET ABBREVIATIONS PER MIL-STD-12

2 INTERPRET ELECTRICAL AND ELECTRONIC REF DESIGNATIONS PER MIL-STD-16

3 REFERENCE DRAWINGS

A TAPE READER ELECTRICAL SCHEMATIC NUMBER L200 003 151

B COMPONENT BOARD ASSY NO L536 001527

C TAPE READER ASS'Y NO L200 003094

Figure 8-3. Tape Reader, Wiring Diagram

Section IX
CARD CHECKER

9-1. DESCRIPTION

During troubleshooting and maintenance, the card checker is used with standard test equipment, including an oscilloscope for monitoring purposes, to provide the required controls and signals for either nominal or marginal checks of each of the component cards. See figure 9-1.

The cards are placed on the front panel connectors of the checker. Cards 10, 13, 14, 18, 20, and 21 use connector 2; all other small cards use connector 1. Placement of the cards on the front panel provides easy access to both sides of the cards for checking of components, inputs, and outputs, which are not available through the connector.

All connector pins are available at test points for monitoring and for applying input signals not available through the card checker switches. The switches control inputs to all connector pins except pins 1 through 7 on connector 1. These pins are the voltage inputs to the circuit cards and are the same for all cards except cards 10, 13, 18, 20, and 21 which require their power inputs to be jumpered through test points. Along with the test points to the connector, additional test points are available to supply signals and test circuits which are used for specific circuit tests. See figures 9-2 through 9-5 for circuit diagrams.

9-2. THEORY OF OPERATION

The following paragraphs discuss the method by which each type of circuit card is tested by the card checker. All circuit checks can be performed with nominal or marginal values in voltage levels and frequency.

9-3. Logic Gates

The card checker provides signals for checking logic gates in static or dynamic modes. These signals are provided by switches which control the inputs to each logic element. The setting of the switches are determined by the logic gate to be checked.

9-4. Flip-flops

The signals for checking a flip-flop are the clock, pull-ups (power to trigger the flip-flop), and a capacitance load. The logic input to the flip-flop is removed and the pull-ups installed. This provides a trigger needed by the flip-flop to operate at clock frequency. The capacitance load is placed at the output of the flip-flop to check the output waveform with a consistant load. Using these connections, the operator can keep the flip-flop in proper operating condition and can troubleshoot the circuit with no great difficulty.

9-5. Emitter Followers

The tests for the emitter followers are the same as the tests for the logic gates.

9-6. Read Flip-Flops

The signals for checking a read flip-flop are the clock, a sine wave in phase with the clock, and a sine wave 180 degrees out of phase with the clock. Added controls of gain and phase shift are provided to obtain marginal conditions of input signals to the read flip-flop. The output of the read flip-flop for a 0 degree sine wave input is a DC level of approximately +35V on the set output and a DC voltage of approximately +9V on the reset output. The output of the read flip-flop for a 180 degree sine wave input is a DC level of approximately +9V on the set output and a DC level of approximately +35V on the reset output.

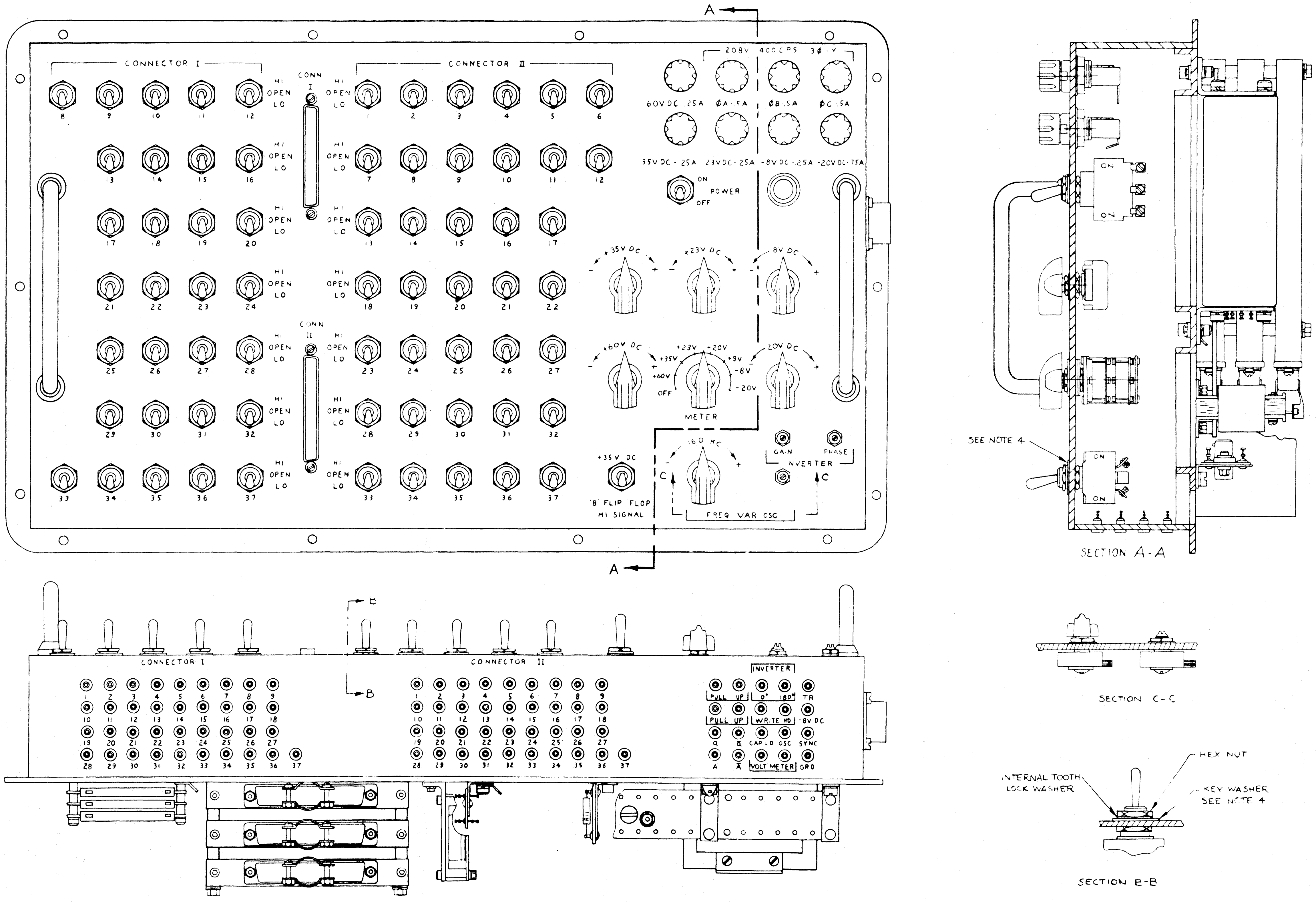


Figure 9-1. Card Checker Control Panel

9-7. Write Amplifiers

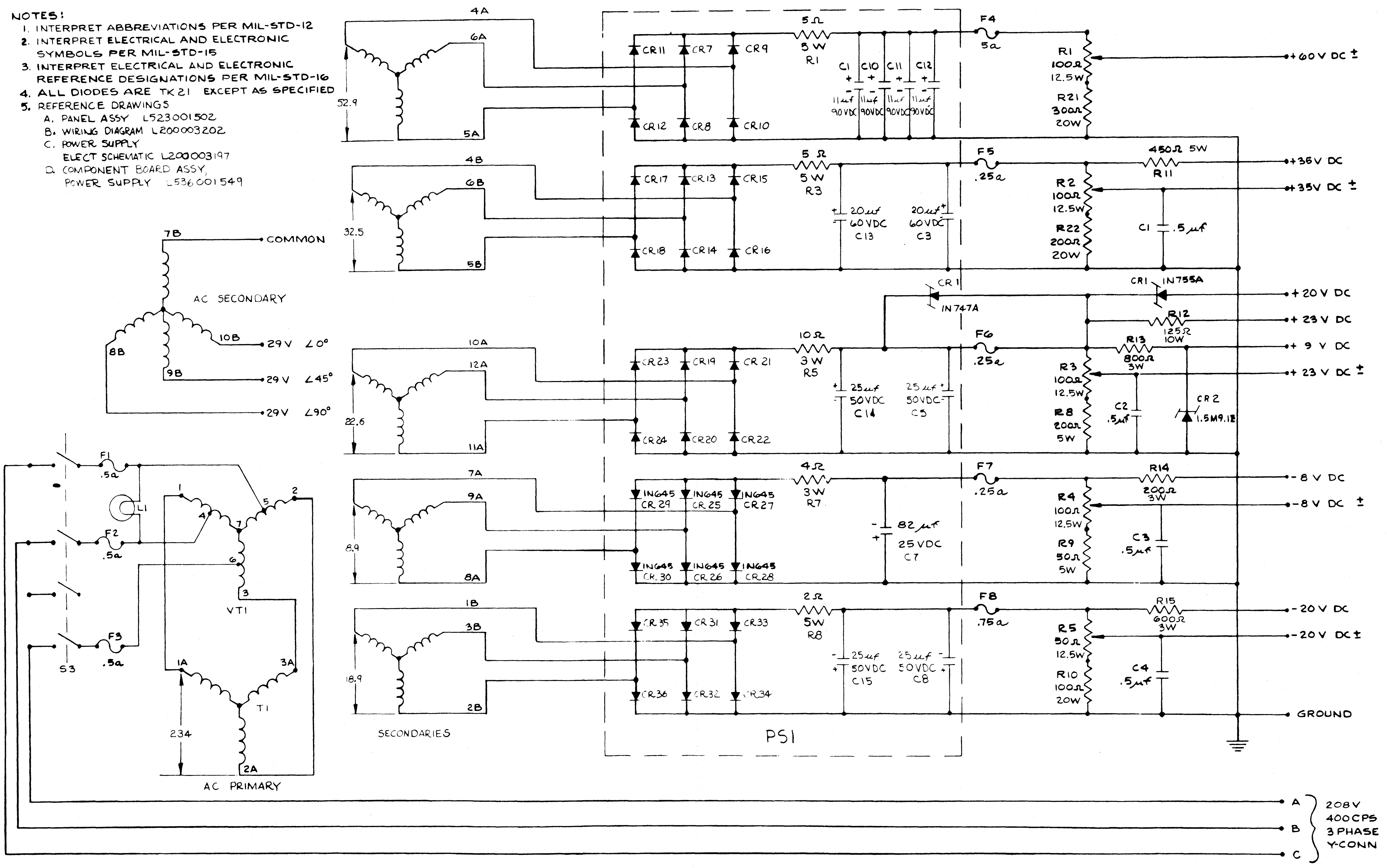
In checking a write amplifier, the card checker provides a clock, a flip-flop for an input signal, and a write head as a load. The flip-flop input and the write head load are available at test points to jumper into the write amplifier and to monitor the input and output signals.

9-8. Clock Generators

The clock generator input signal from the card checker is a 160 KC ± 16 KC sine wave. Both the input and output signals are monitored through test points.

9-9. Head Selection Cards

The head selection cards contain logic gates and may be tested in the same manner as the logic gates are tested. Simulation of head input may be made with a sine wave oscillator. The output may be selected by imposing the required logic conditions on the desired gate.



NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15
3. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
4. ALL DIODES ARE TK 21 EXCEPT AS SPECIFIED
5. REFERENCE DRAWINGS
   A. PANEL ASSY L523001502
   B. WIRING DIAGRAM L200003202
   C. POWER SUPPLY
      ELECT SCHEMATIC L200003197
   D. COMPONENT BOARD ASSY,
      POWER SUPPLY L536001549

Figure 9-2. Power Supply, Schematic Diagram

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15
3. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
4. ALL RESISTOR VALUES ARE EXPRESSED IN OHMS, 1/4 WATT, 5% UNLESS OTHERWISE SPECIFIED
5. REFERENCE DRAWING: COMPONENT BOARD ASSEMBLY, INVERTER AND OSCILLATOR L536 001 535

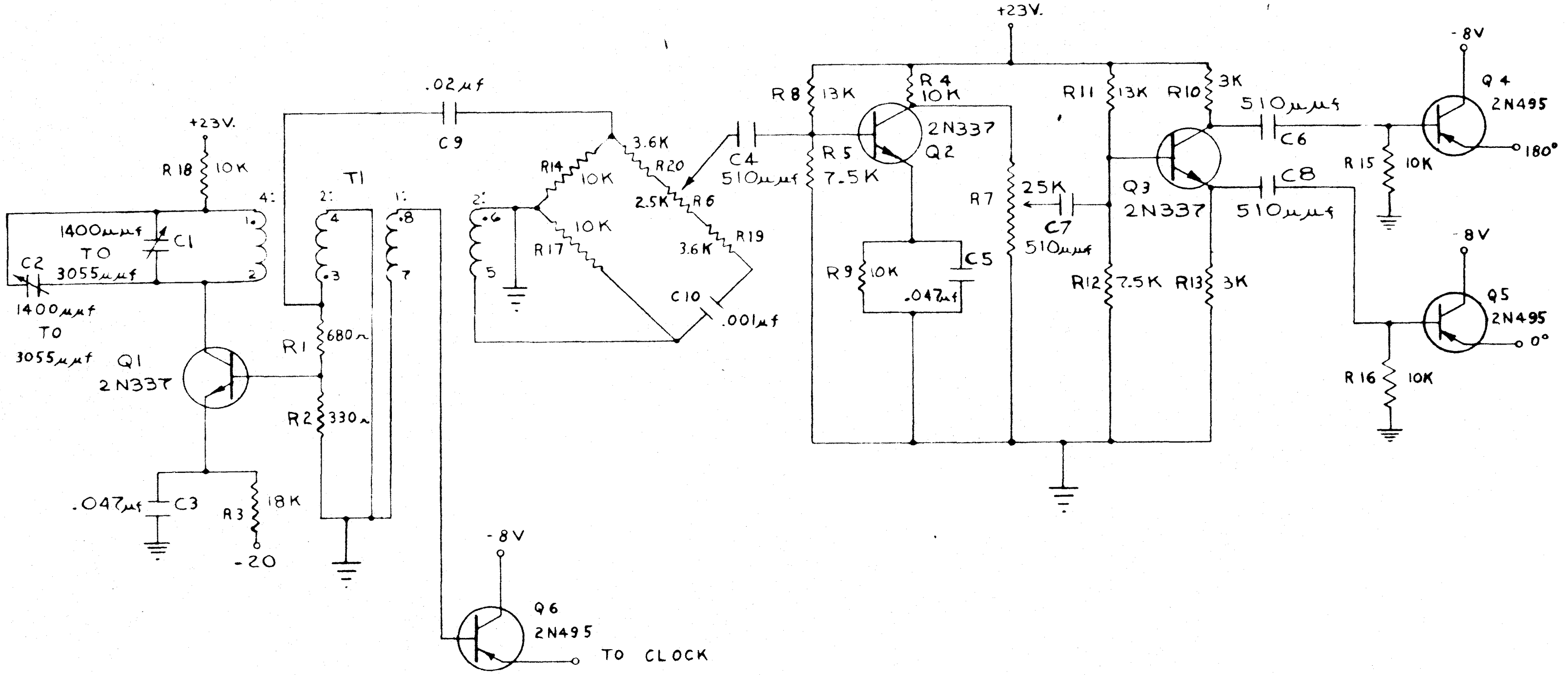
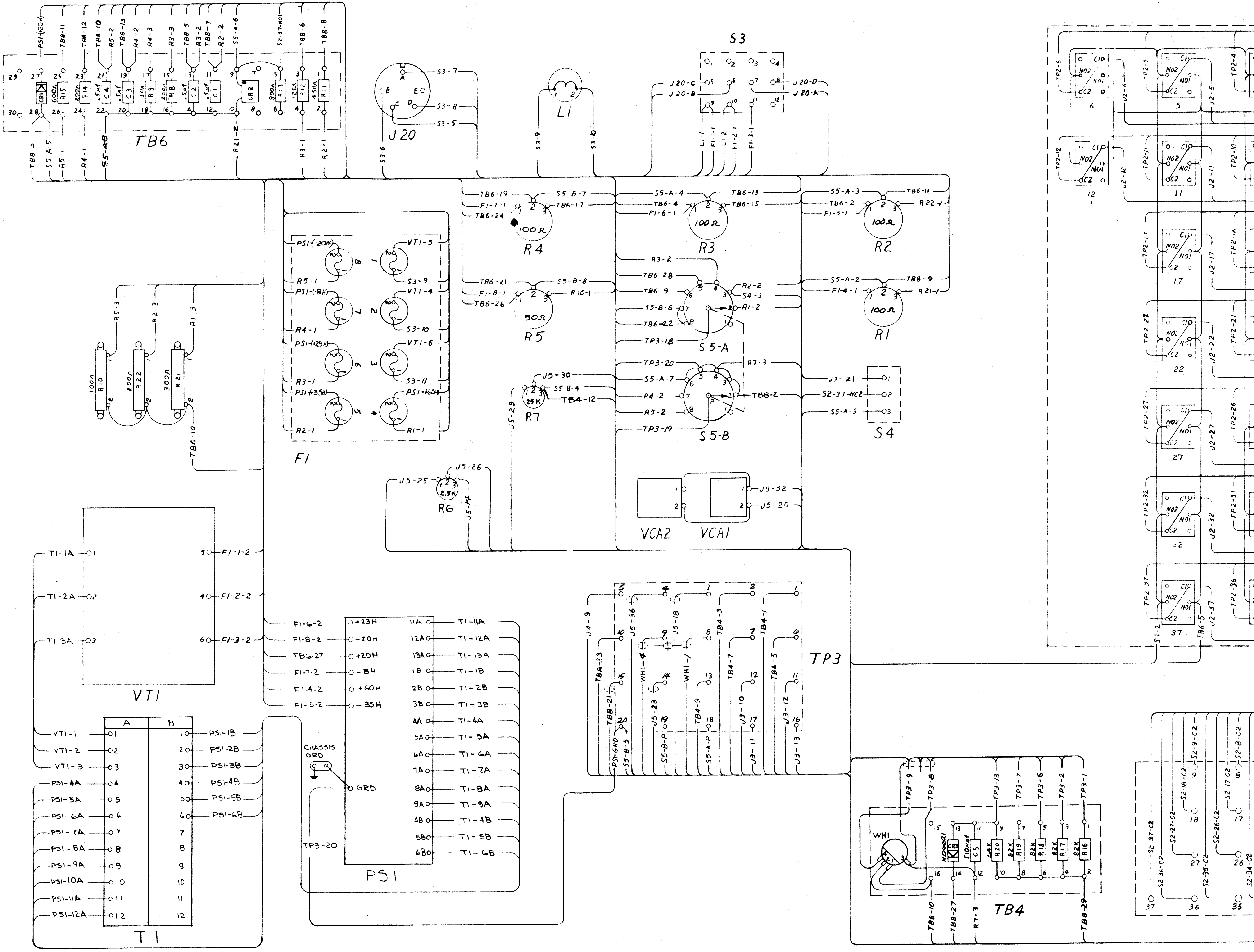
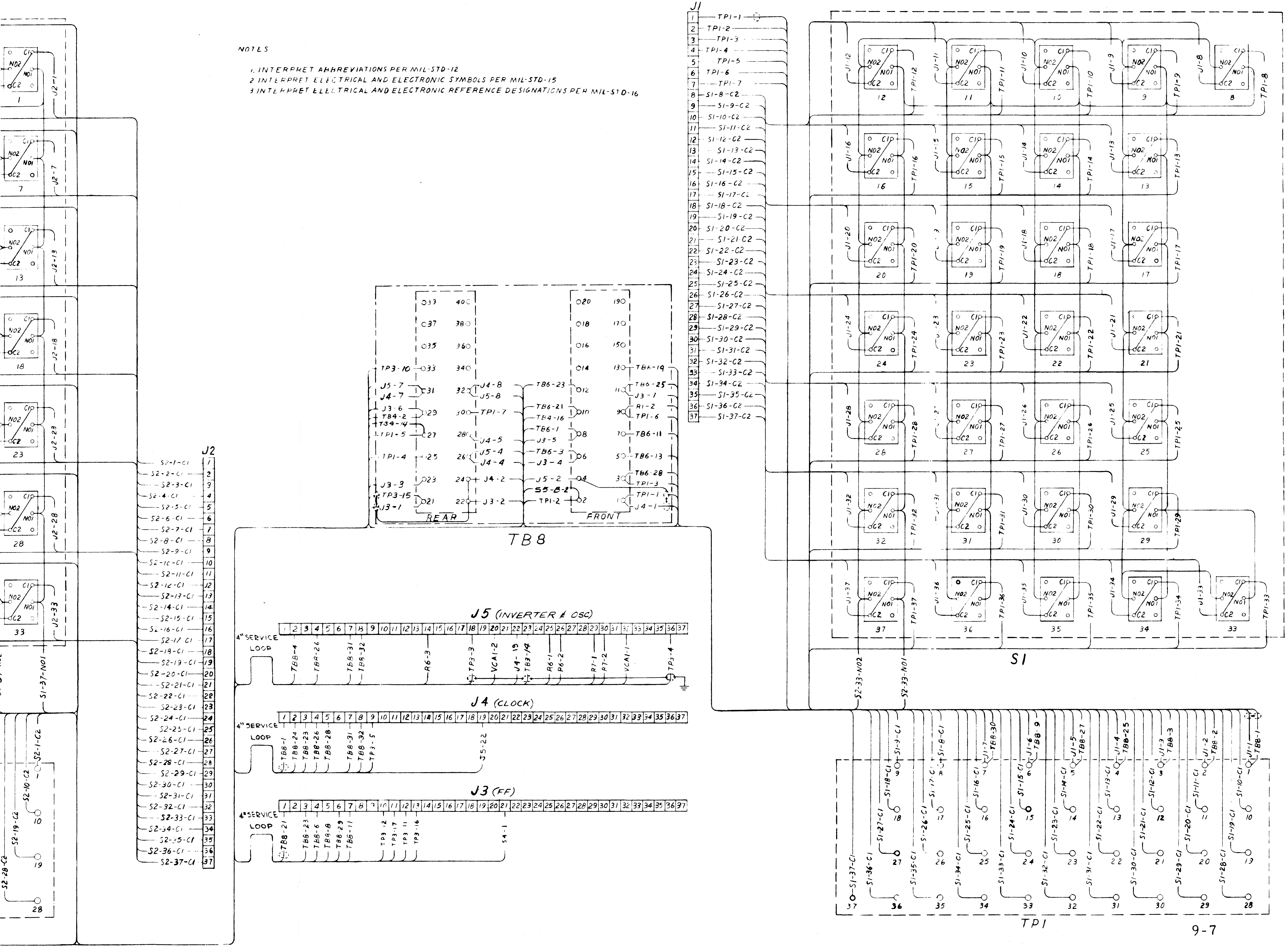


Figure 9-3. Inverter Oscillator, Schematic Diagram


TB6
CR1
R15
R14
C4
C3
R9
R8
C2
C1
CR2
R3
R12
R11
600Ω
200Ω
50Ω
800Ω
125Ω
450Ω
PS1(+20H)
TB8-11
TB8-12
TB8-10
R5-2
TB8-13
R4-2
R4-3
R3-3
TB8-5
R3-2
TB8-7
R2-2
S5-A-6
S2-37-NO1
TB8-6
TB8-8
TB8-3
S5-A-5
R5-1
R4-1
S5-A8
R21-2
R3-1
R2-1
J20
S3-7
S3-8
S3-5
S3-6
L1
S3-9
S3-10
S3
J20-C
J20-B
J20-D
J20-A
L1-1
F1-1-1
L1-2
F1-2-1
F1-3-1
R4
100Ω
TB6-19
F1-7-1
TB6-24
S5-B-7
TB6-17
R3
S5-A-4
TB6-4
F1-6-1
TB6-13
TB6-15
R2
S5-A-3
TB6-2
F1-5-1
TB6-11
R22-1
R5
50Ω
TB6-21
F1-8-1
TB6-26
S5-B-8
R10-1
S5-A
R3-2
TB6-28
TB6-9
S5-B-6
TB6-22
TP3-18
R2-2
S4-3
R1-2
R1
S5-A-2
F1-4-1
TB8-9
R21-1
S5-B
TP3-20
S5-A-7
R4-2
R5-2
TP3-19
R7-3
TB8-2
R7
25K
J5-30
S5-B-4
TB4-12
J5-29
S4
J3-21
S2-37-NC2
S5-A-3
R6
2.5K
J5-25
J5-26
J5-14
VCA2
VCA1
J5-32
J5-20
F1
PS1(-20H)
R5-1
PS1(-8H)
R4-1
PS1(+23H)
R3-1
PS1(+35H)
R2-1
VT1-5
S3-9
VT1-4
S3-10
VT1-6
S3-11
PS1(+60H)
R1-1
R10
R22
R21
100Ω
200Ω
300Ω
R5-3
R2-3
R1-3
TB6-10
VT1
T1-1A
T1-2A
T1-3A
F1-1-2
F1-2-2
F1-3-2
T1
VT1-1
VT1-2
VT1-3
PS1-4A
PS1-5A
PS1-6A
PS1-7A
PS1-8A
PS1-9A
PS1-10A
PS1-11A
PS1-12A
PS1-1B
PS1-2B
PS1-3B
PS1-4B
PS1-5B
PS1-6B
PS1
F1-6-2
F1-8-2
TB6-27
F1-7-2
F1-4-2
F1-5-2
+23H
-20H
+20H
-8H
+60H
-35H
GRD
CHASSIS GRD
TP3-20
T1-11A
T1-12A
T1-13A
T1-1B
T1-2B
T1-3B
T1-4A
T1-5A
T1-6A
T1-7A
T1-8A
T1-9A
T1-4B
T1-5B
T1-6B
TP3
J4-9
J5-36
J5-18
TB4-3
TB4-1
TB8-33
WH1-4
WH1-1
TB4-7
TB4-5
TB8-21
J5-23
TB4-9
J3-10
J3-12
PS1-GRD
S5-B-5
S5-B-P
S5-A-P
J3-11
J3-13
TB4
WH1
HDG621
CR3
510μμf
C5
2.4K
R20
8.2K
R19
R18
R17
R16
TP3-9
TP3-8
TP3-13
TP3-7
TP3-6
TP3-2
TP3-1
TB8-10
TB8-27
R7-3
TB8-29
C1
C2
NO1
NO2
TP2-6
TP2-5
TP2-4
J2-6
J2-5
TP2-12
TP2-11
TP2-10
J2-12
J2-11
TP2-17
TP2-16
J2-17
TP2-22
TP2-21
J2-22
TP2-27
TP2-26
J2-27
TP2-32
TP2-31
J2-32
TP2-37
TP2-36
J2-37
S1-2
TB6-5
S2-9-C2
S2-8-C2
S2-18-C2
S2-17-C2
S2-37-C2
S2-27-C2
S2-26-C2
S2-36-C2
S2-35-C2
S2-34-C2



NOTES

1. INTERPRET ABBREVIATIONS PER MIL-STD-12
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15
3. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16

Figure 9-4.
Card Checker, Wiring Diagram

NOTES:

1. INTERPRET ABBREVIATIONS PER MIL-STD-12
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15
3. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
4. ALL RESISTOR VALUES ARE EXPRESSED IN OHMS, 1/4 WATT, 5% UNLESS OTHERWISE SPECIFIED
5. REFERENCE DRAWING: COMPONENT BOARD ASSY, L536 001541

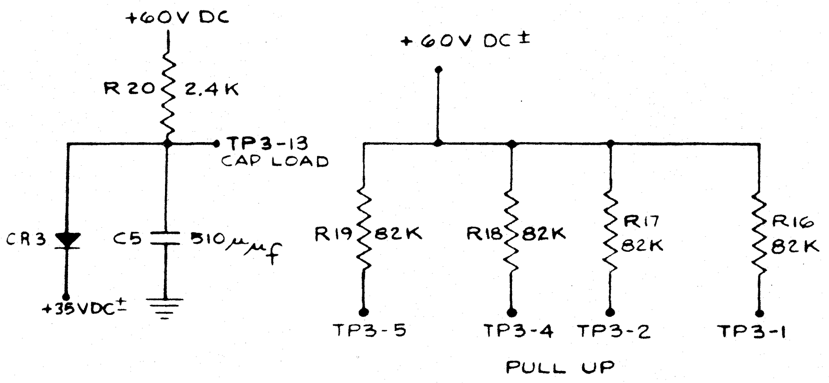


Figure 9-5. Test Circuits (FF), Schematic Diagram

Appendix A
WIRING TABLES

The following tables contain wiring information pertaining to the computer's external connectors (4J1, 4J2, 4J3, and 4J4) and the connectors within the memory (P21, P23, P25, and P26).

CONNECTOR 4J1

Pin	Signal		Pin	Signal
A	115 Volts 400 cps T_1		Z	R_3
B	Spare - Do Not Use		a	$\overline{F}_p$
C	115 Volts - 400 cps T_2		b	$\overline{F}_n$
D	115 Volts - 400 cps T_3		c	$+\Delta V_w$
E	Spare - Do Not Use		d	$-\Delta V_w$
F	Power Gnd. (Neutral)		e	F_{r8}
G	Spare		f	F_{r4}
H	Spare		g	Δt_1 (1.3 kc)
J	+60V DC		h	+60 I/O
K	+35V DC		i	+35 I/O
L	K_4		j	+23 I/O
M	K_5		k	+20 I/O
N	$\overline{K}_5$		m	-20 I/O
P	$\overline{K}_6$		n	S_1
R	$\overline{F}_0$		p	$\overline{S}_1$
S	+20V DC		q	S_2
T	$\overline{R}_3$		r	$\overline{S}_2$
U	+23V DC		s	$\overline{S}_3$
V	Spare	Do Not Use	t	S_4
W	Spare		u	$\overline{S}_4$
X	Spare		v	Y_{02}
Y	Spare		w	W_2

CONNECTOR 4J1 (Cont.)

Pin	Signal	Pin	Signal
x	$\overline{W}_3$	CC	Computer Clock (T)
y	P_{24}	DD	-8V DC
z	$\overline{H}$	EE	-20V DC
AA	S_6	FF	P_0
		GG	+35 Volts W/A
BB	$\overline{S_6}$	HH	Computer Gnd.

CONNECTOR 4J2

Pin	Signal	Pin	Signal
A	B_0	a	B_8*
B	B_1	b	B_9*
C	B_2	c	B_{10}*
D	B_3	d	B_{11}*
E	B_4	e	B_{12}*
F	B_5	f	B_{13}*
G	B_6	g	B_{14}*
H	B_7	h	B_{15}*
J	B_8	i	$\overline{W}_2$
K	B_9	j	W_3
L	B_{10}	k	W_4
M	B_{11}	m	$\overline{W}_4$
N	B_{12}	n	W_5
P	B_{13}	p	$\overline{W}_5$
R	B_{14}	q	$-\Delta V_v$
S	B_0*	r	$+\Delta V_v$
T	B_1*	s	Δt_2
U	B_2*	t	$\overline{A}_1$
V	B_3*	u	$\overline{K}_4$
W	B_4*	v	K_6
X	B_5*	w	Y_{01}
Y	B_6*	x	$\overline{Y}_{02}$
Z	B_7*	y	W_1

CONNECTOR 4J2 (Cont.)

Pin	Signal	Pin	Signal
z	$\overline{W}_1$	EE	R_4
AA	$K_1 K_2 K_3$	FF	$+\Delta V_u$
BB	S_3		
CC	F_g	GG	$- \Delta V_u$
DD	R_1	HH	H

CONNECTOR 4J3

Pin	Signal	Pin	Signal
A	A_1	c	$\overline{F}_1$
B	K_5	d	F_c
C	0_3	e	$\overline{M}_{0x}$
D	$\overline{X}_{01}$	f	$\overline{C}_5$
E	$\overline{K}_5$	g	$\overline{F}_c$
F	$\overline{0}_3$	h	$\overline{X}_{02}$
G	$\overline{F}_a$	i	X_{02}
H	F_f	j	Y_{01}
J	F_a	k	$\overline{0}_2$
K	P_o	m	$\overline{K}_4$
L	$\overline{P}_o$	n	F_b
M	F_g	p	T_1
N	Y_{02}	q	M_d
P	D_1	r	0_2
R	$\overline{D}_0$	s	K_4
S	E_3	t	T_4
T	$\overline{D}_1$	u	$\overline{0}_1$
U	R_0	v	$\overline{K}_3$
V	F_j	w	T_5
W	D_0	x	K_c
X	$\overline{Y}_0$	y	F
Y	$\overline{G}$	z	K_1
Z	$\overline{T}_3$	AA	0_1
a	$\overline{T}_1$	BB	$\overline{E}_3$
b	F_3	CC	R_0

CONNECTOR 4J3 (Cont.)

Pin	Signal	Pin	Signal
DD	M_{0y}	FF	T_3
		GG	F_d
EE	M_m	HH	$\overline{F}_2$

CONNECTOR 4J4

Pin	Signal	Pin	Signal
A	F_1	e	$\overline{I}_0$
B	$\overline{E}_8$	f	L
C	F_h	g	$\overline{P}_5$
D	C_5	h	C_4
E	T_s	i	P_5
F	$\overline{C}_4$	j	F_m
G	M_{0x}	k	$\overline{T}_5$
H	F_t	m	$\overline{S}_{0T}$
J	J	n	$\overline{T}_4$
K	$\overline{A}_1$	p	$\overline{C}_a$
L	A_0	q	$\overline{K}_Z$
M	$\overline{P}_{24}$	r	$\overline{K}_6$
N	C_a	s	$\overline{K}_1$
P	$\overline{A}_0$	t	$\overline{T}_2$
R	$\overline{J}$	u	$\overline{T}_5$
S	K_6	v	K_2
T	$\overline{S}_5$	w	P_{24}
U	K_3	x	S_{0T}
V	X_{01}	y	Signal Ground
W	$K_1\overline{K}_2$	z	I_0
X	S_5	AA	T_2
Y	$\overline{M}_m$	BB	F_2
Z	$\overline{K}_c$	CC	E_f
a	$\overline{Y}_{02}$	DD	$\overline{F}_m$
b	$\overline{M}_{0y}$	EE	F_w
		FF	$\overline{F}_w$
c	$\overline{K}_1\overline{K}_2$	GG	Clock Pulse (T)
d	G	HH	$\overline{L}$

CONNECTOR P21

Pin	Signal	Pin	Signal
A	H_0	M	H_{10}
B	H_1	N	H_{11}
C	H_2	P	H_{12}
D	H_3	R	H_{13}
E	H_4	S	H_{14}
F	H_5	T	H_{15}
H	H_6	U	H_0
J	H_7	V	$H_{\overline{\underline{1}}}$
K	H_8	W	-8 V Hd*
L	H_9	X	Shield GND

CONNECTOR P23

Pin	Signal	Pin	Signal
A	H_{16}	M	H_{26}
B	H_{17}	N	H_{27}
C	H_{18}	P	H_{28}
D	H_{19}	R	H_{29}
E	H_{20}	S	H_{30}
F	H_{21}	T	H_{31}
H	H_{22}	U	H_{16}
J	H_{23}	V	$H_{\overline{\underline{17}}}$
K	H_{24}	W	-8 V Hd*
L	H_{25}	X	Shield GND

CONNECTOR P25

Pin	Signal	Pin	Signal
A	D_0	J	H_2
B	S_{ot}	K	M_{0x}
C	X_{01}	L	$\Delta 2$
D	I_o	M	H_3
E	t	N	$\Delta\overline{3}$
F	$\Delta 1$	P	R_0
H	Y_{01}	R	Y_{02}

CONNECTOR P25 (Cont.)

Pin	Signal	Pin	Signal
S	ΔT	V	H_{19}
T	A_0	W	-8 V $\overline{\text{Hd}}$*
U	$H_{\underline{18}}$	X	Shield GND.

CONNECTOR P26

Pin	Signal	Current is Recording
A	ζ	0
B	ζ	1
C	d_o'	0
D	d_o	1
E	x_o'	0
F	x_o	1
H	y_o'	0
J	y_o	1
K	m_x'	0
L	m_x	1
M	a_o'	0
N	a_o	1
P	n_o'	0
R	n_o	1
S	m_v'	0
T	m_v	1
U	m_z'	0
V	m_z	1
W	-20 V Hd	
X	Shield GND	

Appendix B
ADDITIONAL ILLUSTRATIONS

NOTES:
1. ALL RESISTORS UNLESS NOTED ARE CB ¼ W-5%
2. ALL DIODES UNLESS NOTED ARE HD 6621
3. ASSEMBLY DWG NO L536001511

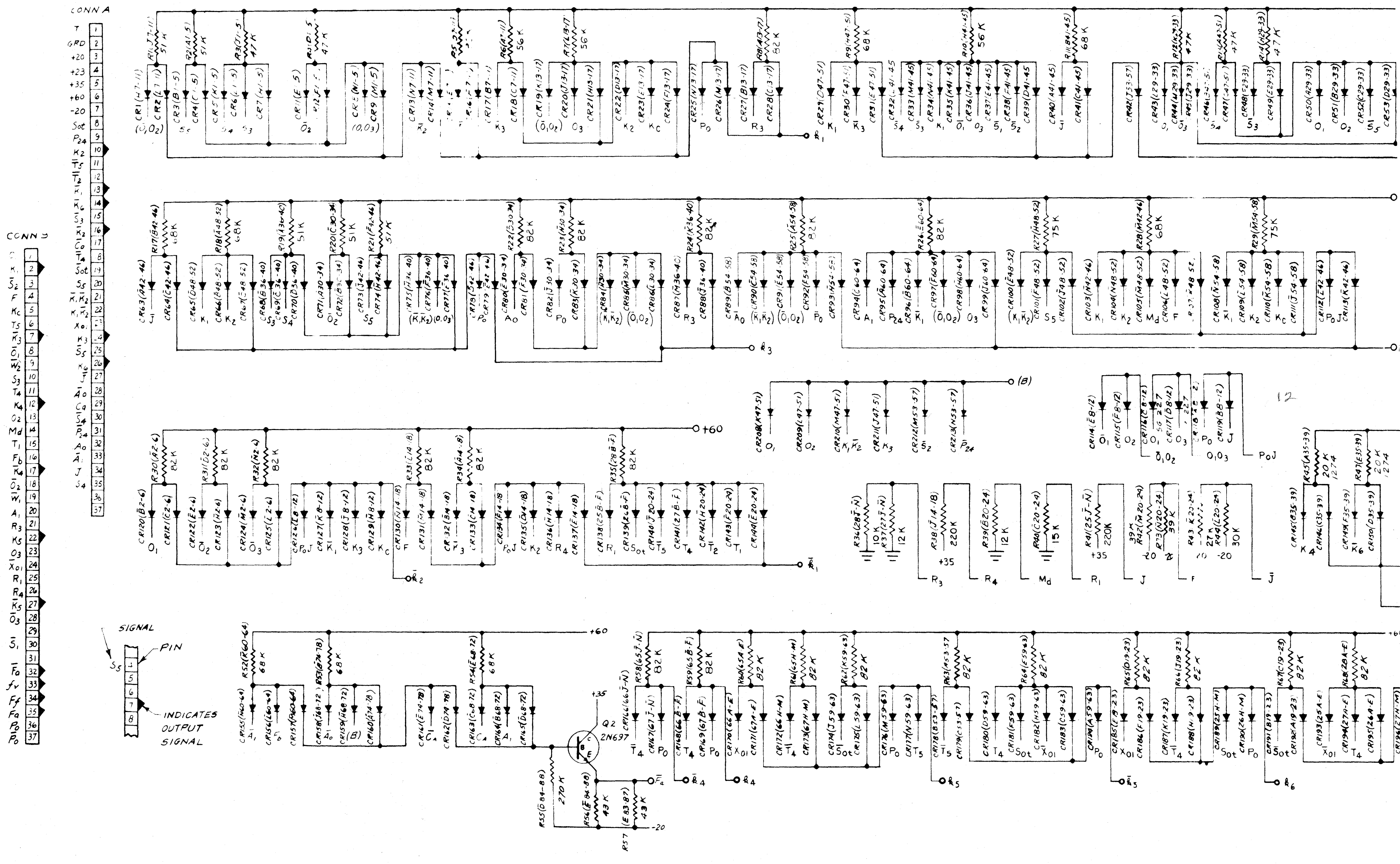


Figure B-1.

NOTES:
1. ALL RESISTORS UNLESS NOTED ARE CB¼W-5%
2. ALL DIODES UNLESS NOTED ARE HD 6621
3 ASSEMBLY DWG NO L536001511

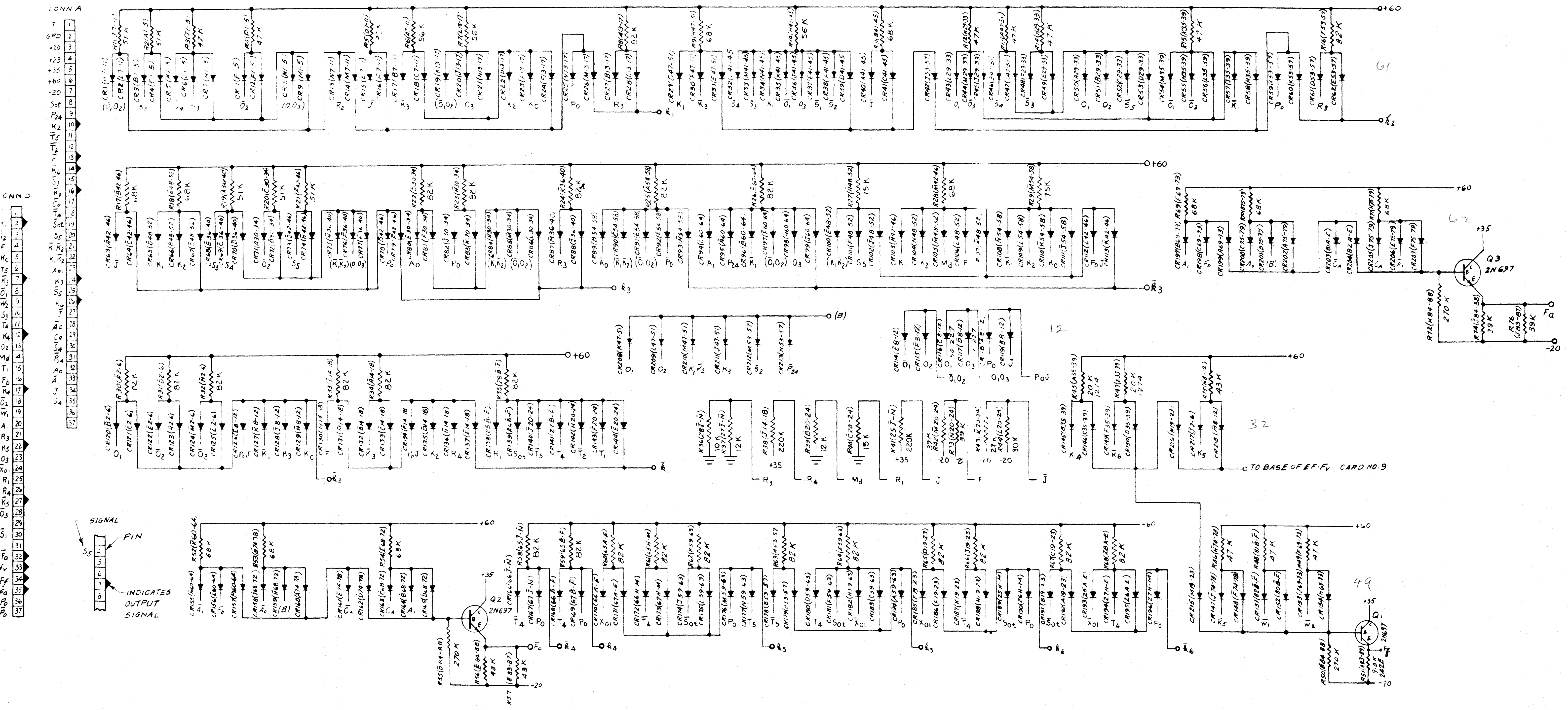
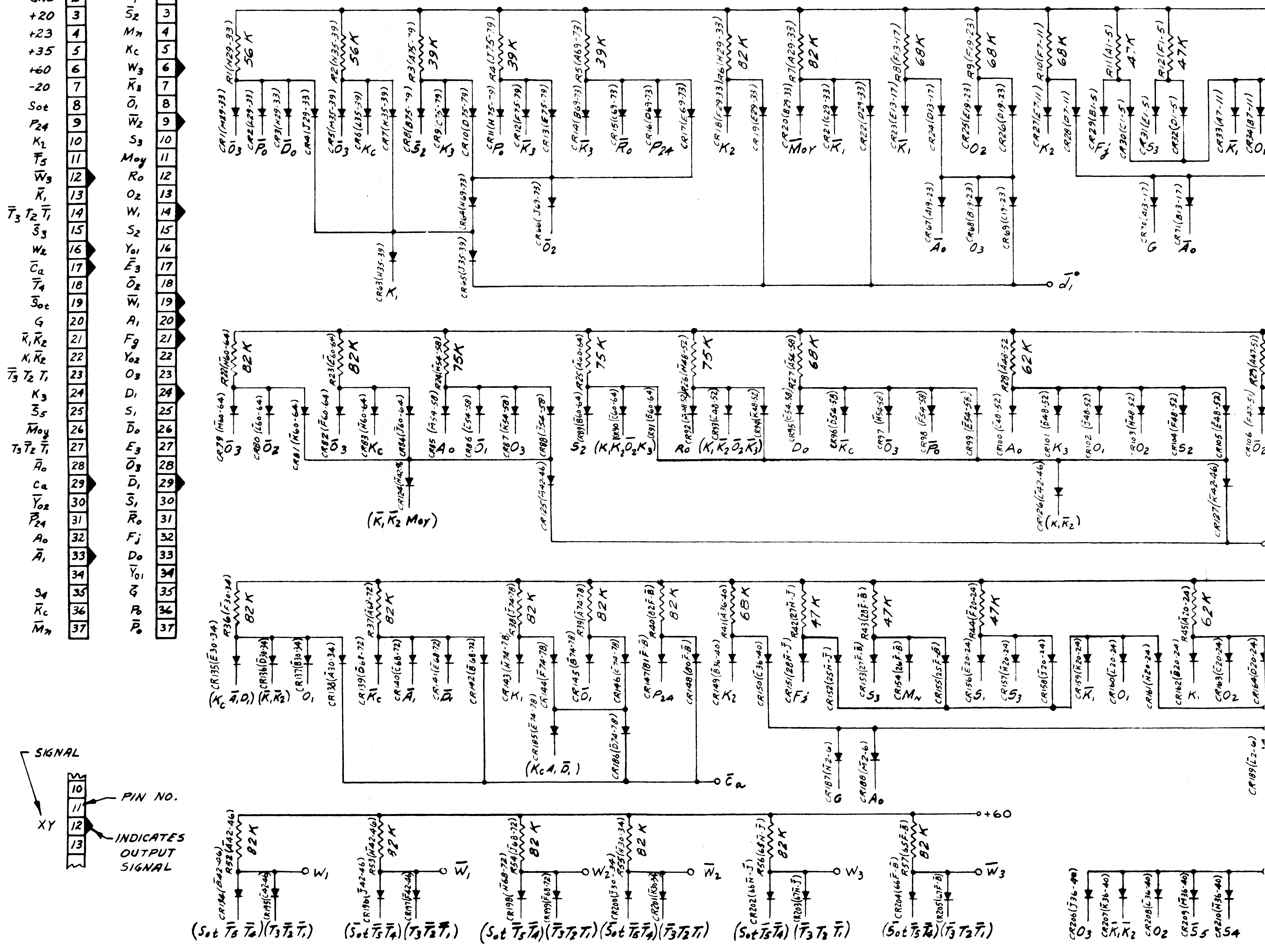


Figure B-1. Logic, Card No. 1, Schematic Diagram

NOTES:

1- ALL RESISTORS, UNLESS NOTED ARE TYPE CB$\frac{1}{4}$ W-5%

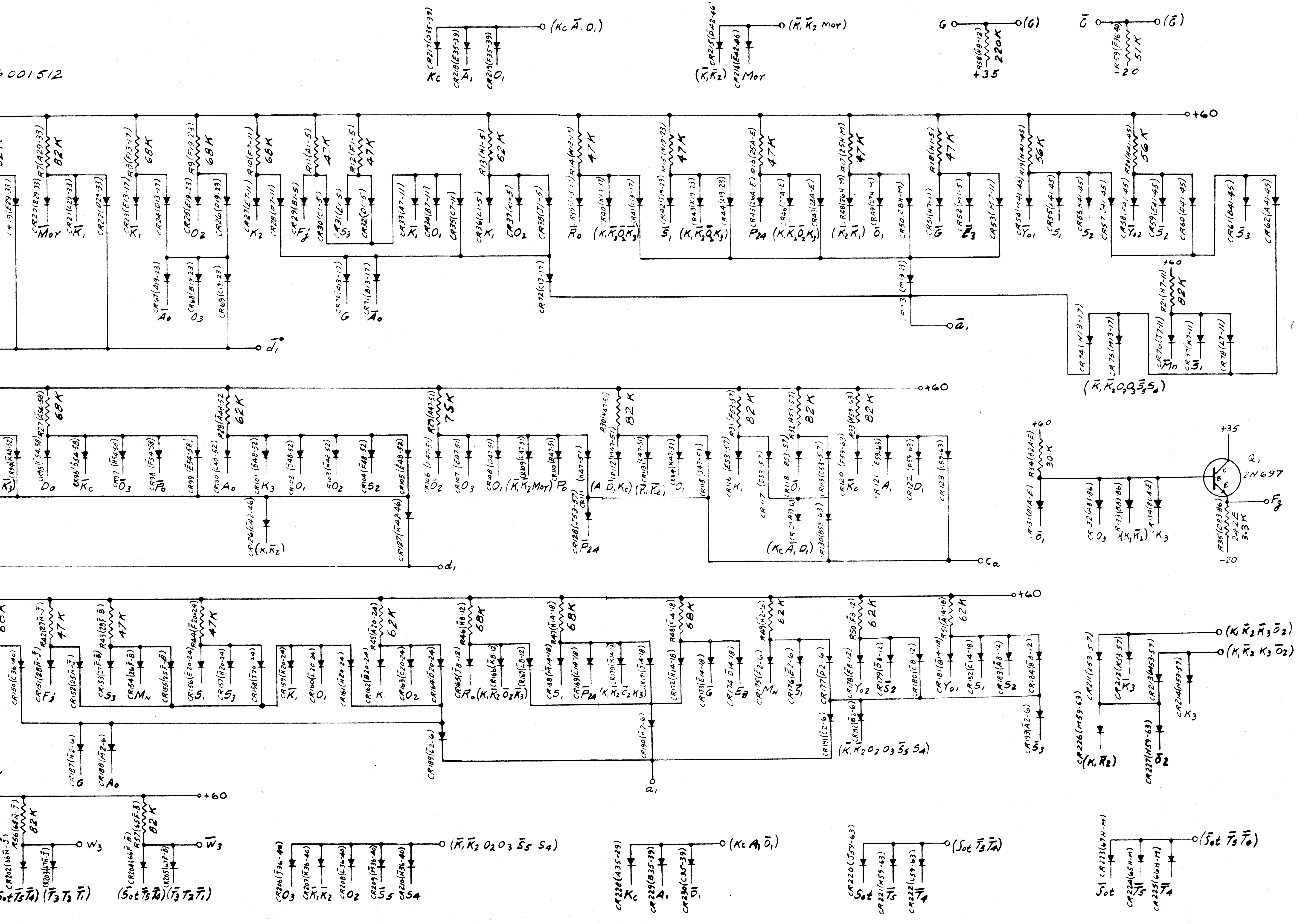
2- FOR ASSEMBLY SEE DWG NO. L536 001 512



CONN. A		CONN. B	
T	1	O_1	1
GND	2	K_1	2
+20	3	$\bar{S}_2$	3
+23	4	M_n	4
+35	5	K_c	5
+60	6	W_3	6 ▶
-20	7	$\bar{K}_3$	7
S_{ot}	8	$\bar{O}_1$	8
P_{24}	9	$\bar{W}_2$	9 ▶
K_2	10	S_3	10
$\bar{T}_5$	11	M_{oy}	11
$\bar{W}_3$	12 ▶	R_o	12
$\bar{K}_1$	13	O_2	13
$\bar{T}_3 T_2 \bar{T}_1$	14	W_1	14 ▶
$\bar{S}_3$	15	S_2	15
W_2	16 ▶	Y_{01}	16
$\bar{C}_a$	17 ▶	$\bar{E}_3$	17
$\bar{T}_4$	18	$\bar{O}_2$	18
$\bar{S}_{ot}$	19	$\bar{W}_1$	19 ▶
G	20	A_1	20 ▶
$\bar{K}_1\bar{K}_2$	21	F_g	21 ▶
$K_1\bar{K}_2$	22	Y_{02}	22
$\bar{T}_3 T_2 T_1$	23	O_3	23
K_3	24	D_1	24 ▶
$\bar{S}_5$	25	S_1	25
$\bar{M}_{oy}$	26	$\bar{D}_o$	26
$T_3\bar{T}_2\bar{T}_1$	27	E_3	27
$\bar{A}_o$	28	$\bar{O}_3$	28
C_a	29 ▶	$\bar{D}_1$	29 ▶
$\bar{Y}_{02}$	30	$\bar{S}_1$	30
$\bar{P}_{24}$	31	$\bar{R}_o$	31
A_o	32	F_j	32
$\bar{A}_1$	33 ▶	D_o	33
	34	$\bar{Y}_{01}$	34
S_4	35	$\bar{G}$	35
$\bar{K}_c$	36	P_o	36
$\bar{M}_n$	37	$\bar{P}_o$	37

Figure B-2. Logic, Card No. 2, Schematic Diagram

6001512



67

16

57

64

26

230

34

NOTES:
1- ALL RESISTORS, UNLESS NOTED, ARE CB ¼ W-5%
2- ASSEMBLY DWG. NO. L536001513.
3- ALL DIODES, UNLESS NOTED, ARE HD 6621.

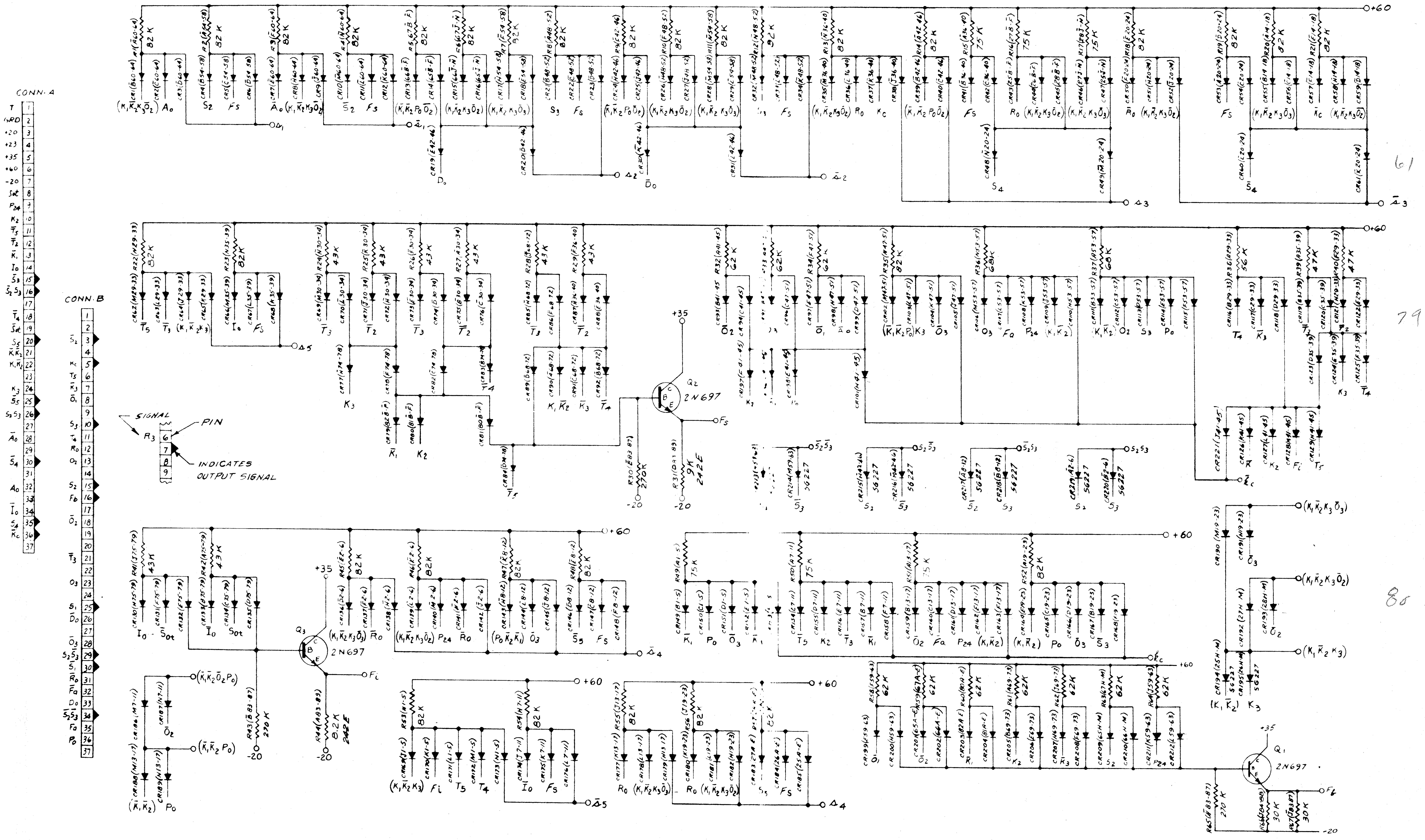


Figure B-3. Logic, Card No. 3, Schematic Diagram

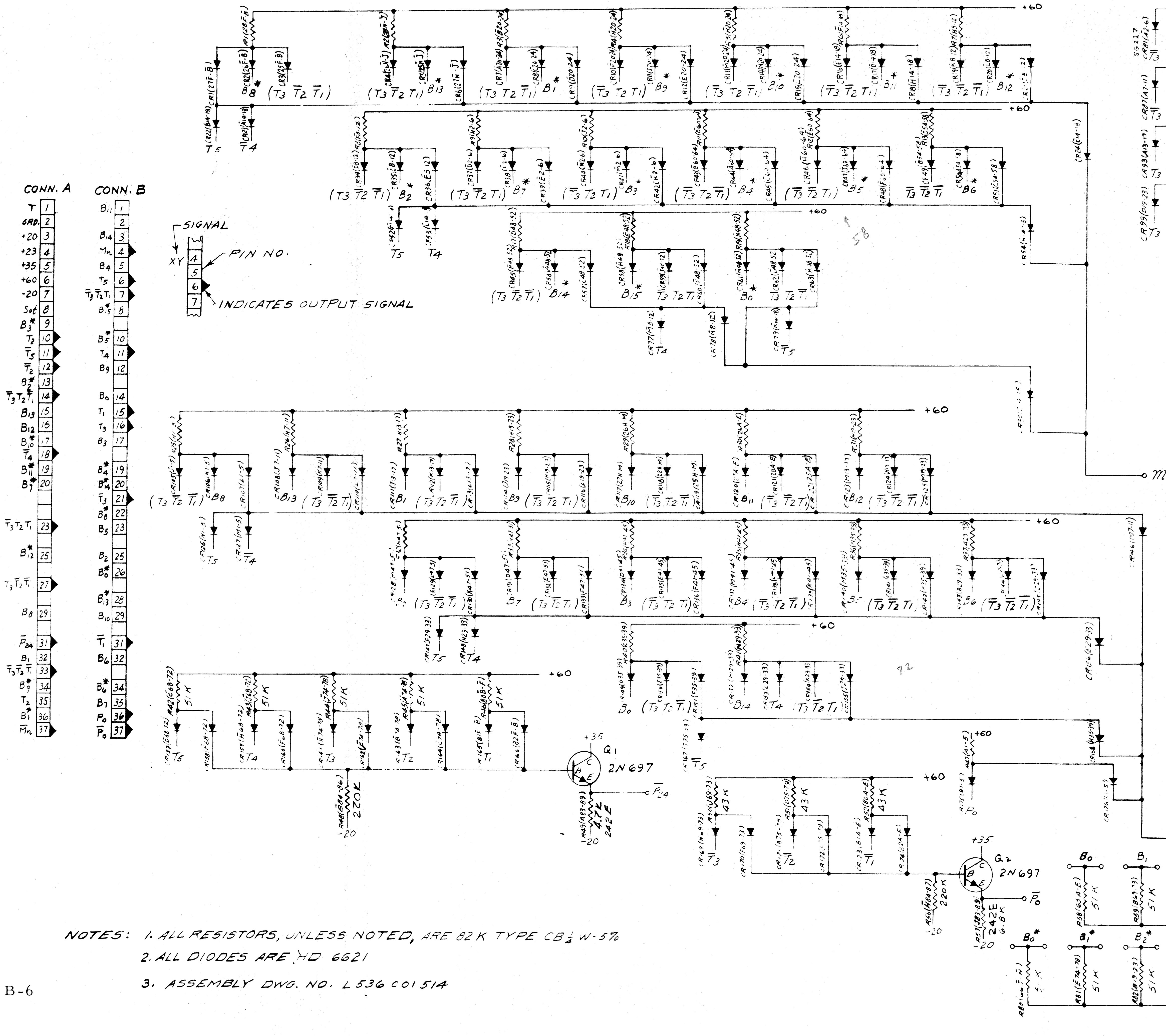
61

79

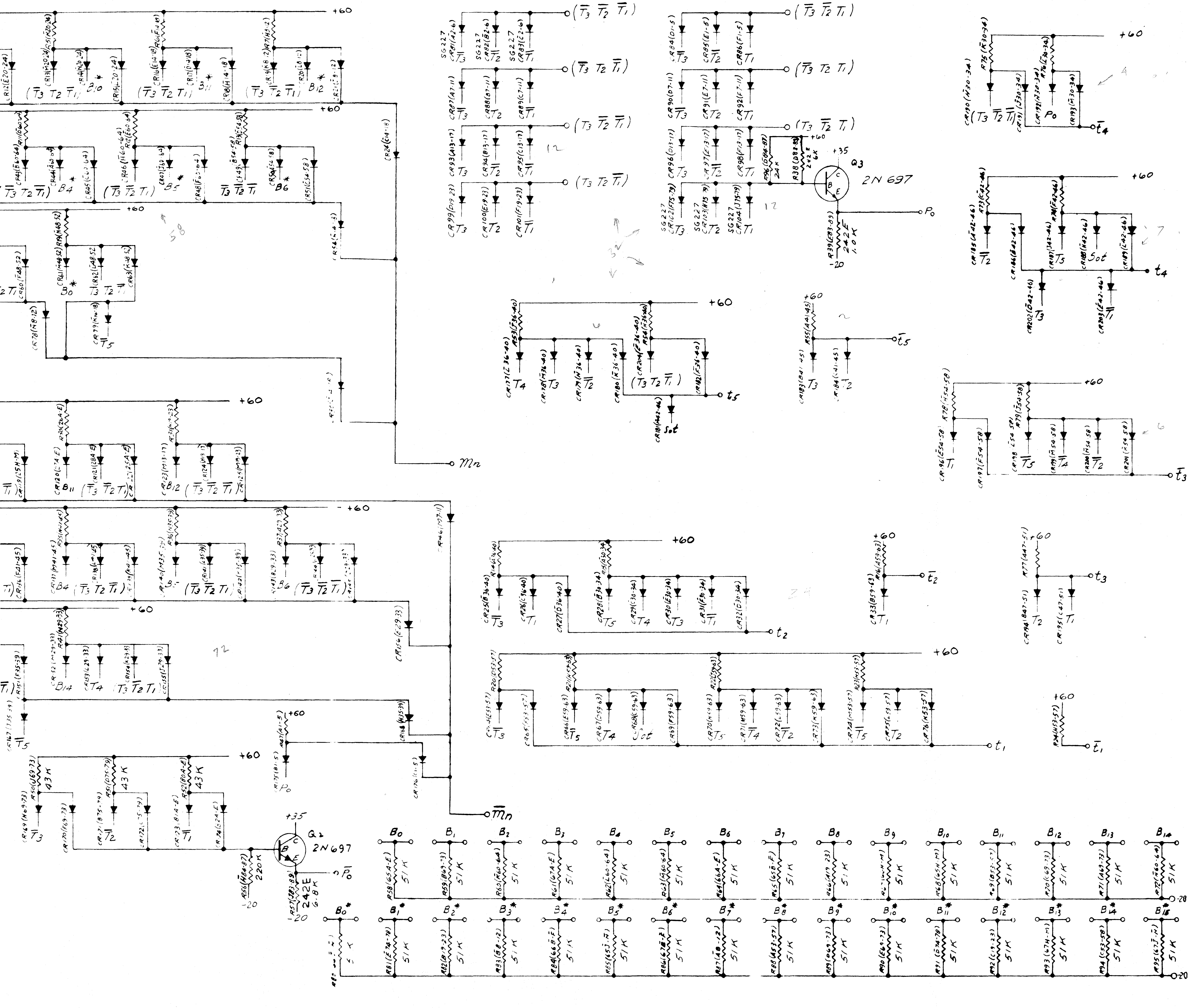
80

220

Figure B-4. Logic, Card No. 4, Schematic Diagram



NOTES: 1. ALL RESISTORS, UNLESS NOTED, ARE 82 K TYPE CB ¼ W - 5%
2. ALL DIODES ARE HD 6621
3. ASSEMBLY DWG. NO. L 536 C01 514


+60
(T̄3 T̄2 T̄1)
(T̄3 T2 T̄1)
(T̄3 T̄2 T1)
(T3 T̄2 T̄1)
(T3 T2 T̄1)
(T3 T̄2 T1)
SG227
B10 *
B11 *
B12 *
B4 *
B5 *
B6 *
B0 *
T̄5
Mn
m̄n
+35
Q3
2N697
P0
R39 (ĒB3-89) 242E 1.0K
-20
R38 (D83-89) 242E 6K
R96 (ĒB4-87) 2A K
t4
Sot
t5
t̄5
t3
t̄3
t2
t̄2
t1
t̄1
B11
B12
B4
B6
B14
T4
Q2
2N697
R56 (ĀEB-87) 220 K
R57 (ĀB-89) 242E 6.8K
R50 (J69-73) 43 K
R51 (D75-79) 43 K
R52 (B0A-E) 43 K
B0
B1
B2
B3
B4
B5
B6
B7
B8
B9
B10
B11
B12
B13
B14
51 K
B0*
B1*
B2*
B3*
B4*
B5*
B6*
B7*
B8*
B9*
B10*
B11*
B12*
B13*
B14*
B15*
-20

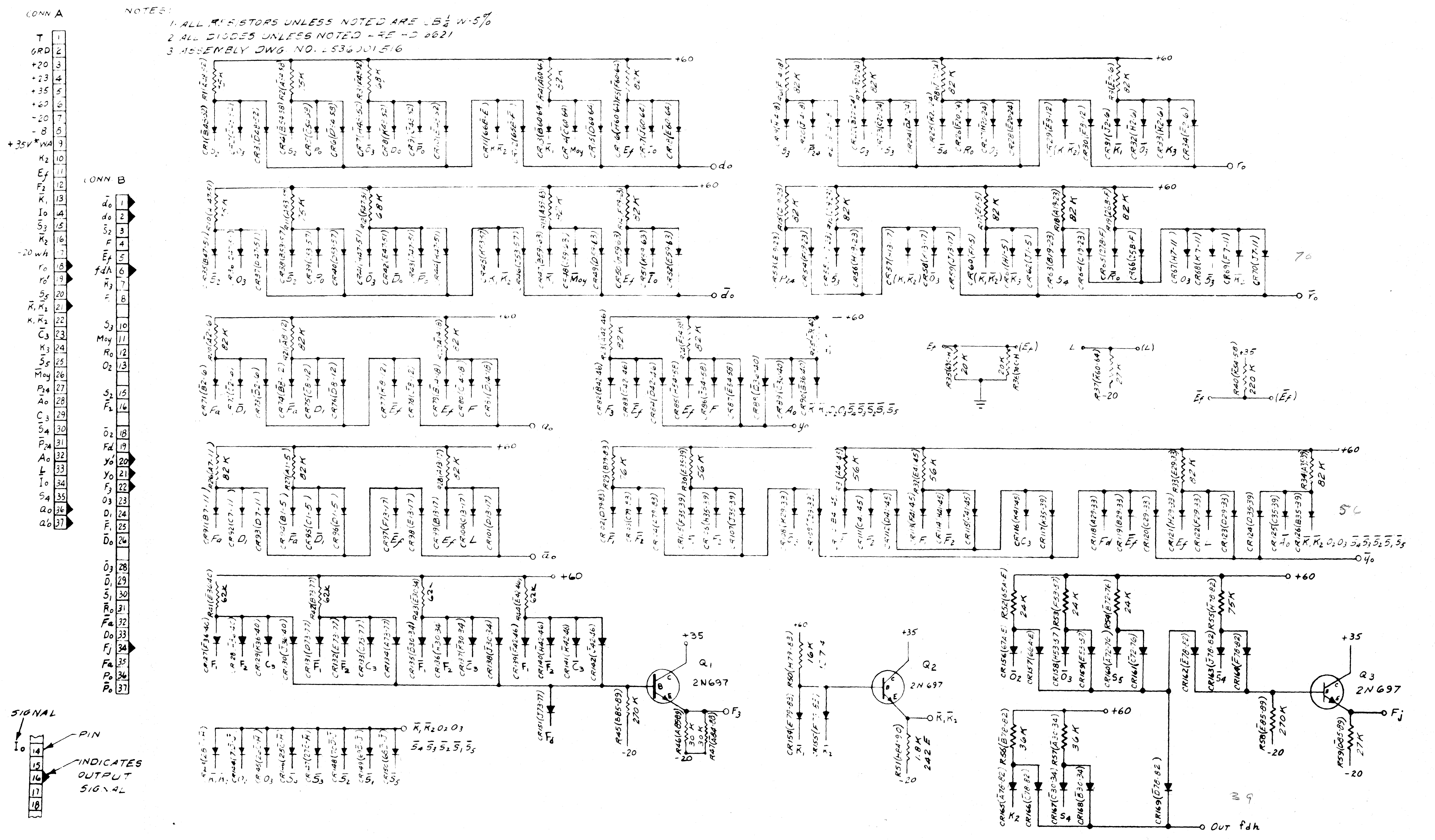


Figure B-5. Logic, Card No. 5, Schematic Diagram

165

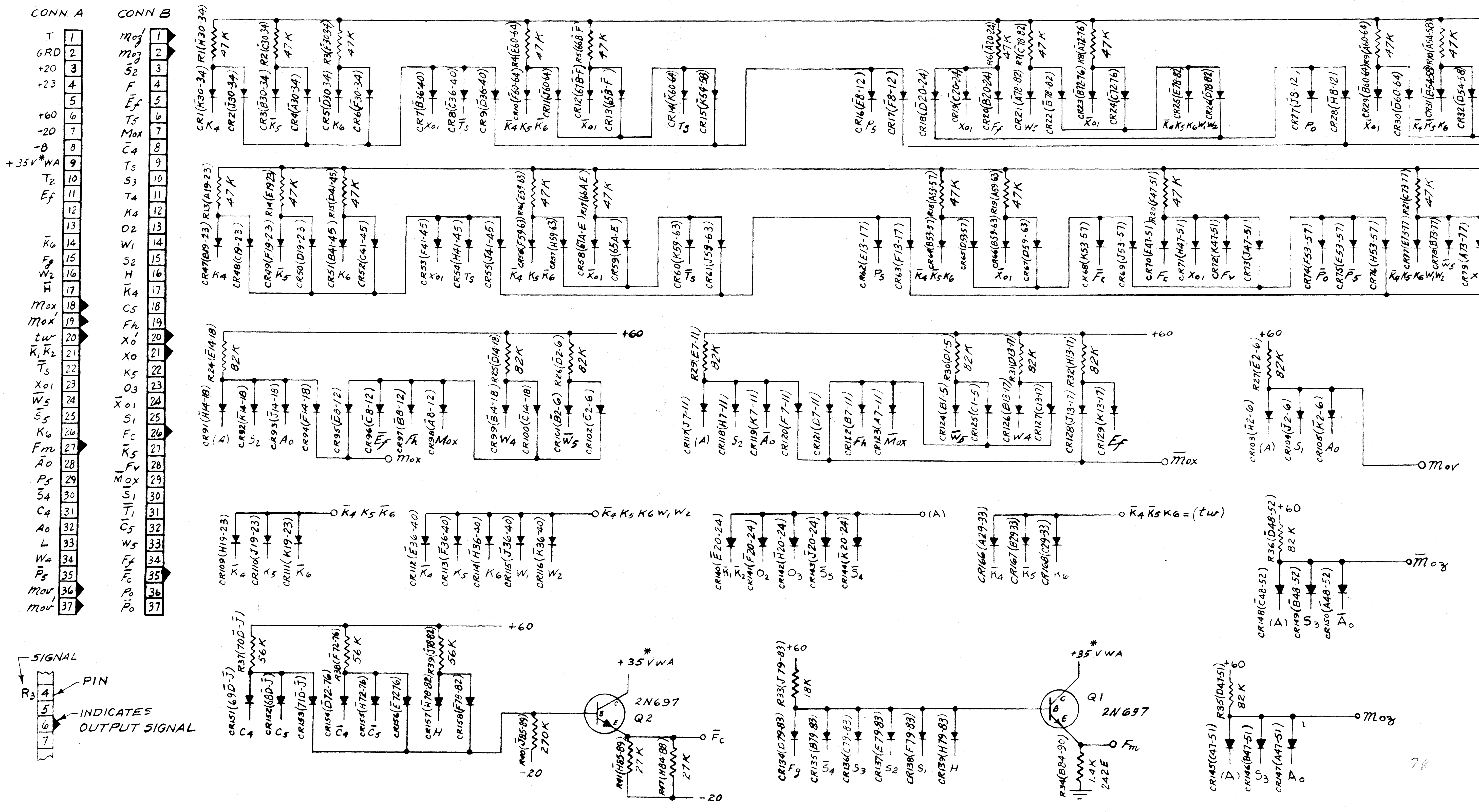
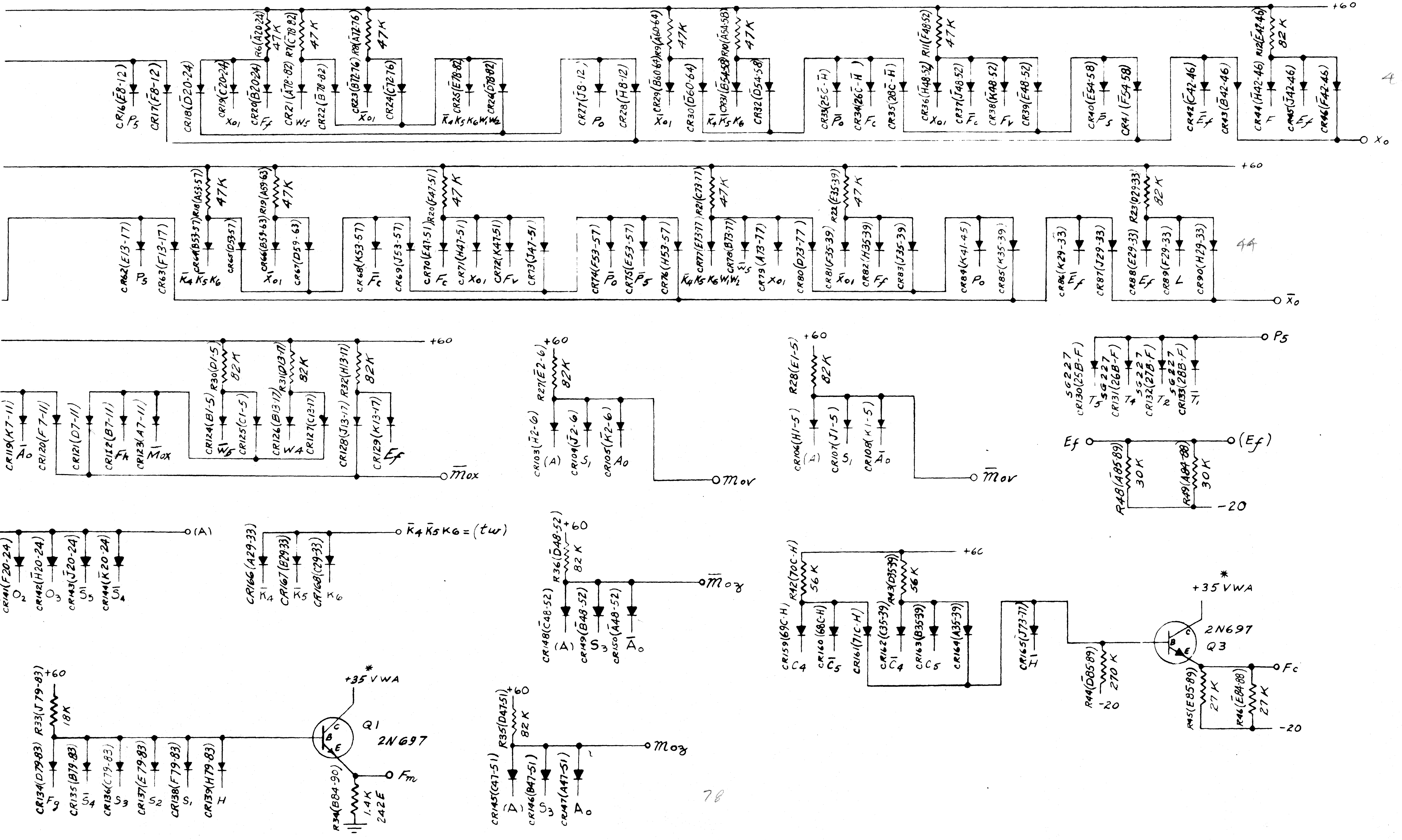


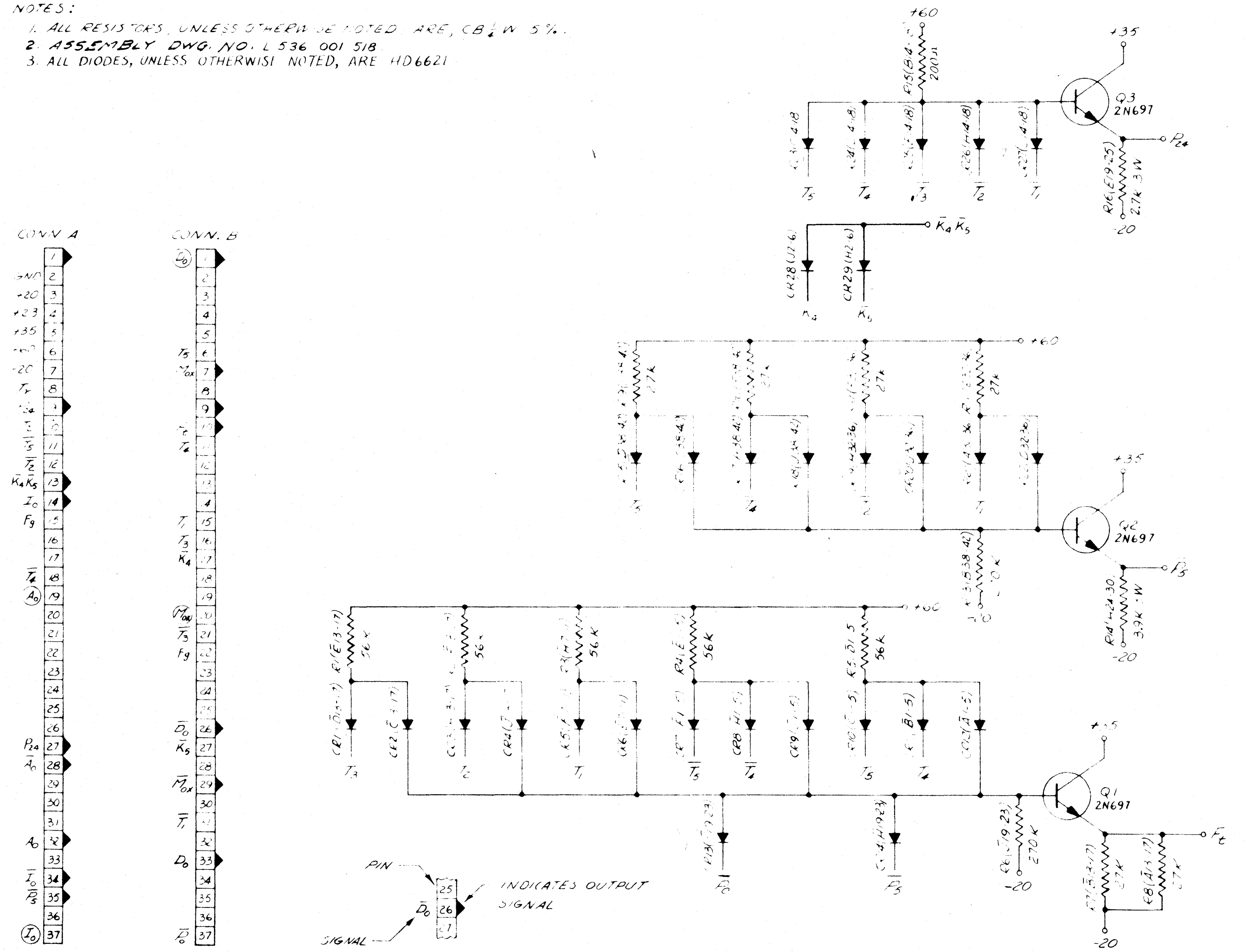
Figure B-6. Logic, Card No. 6, Schematic Diagram

78



+60
R6(Ā20-24) 47K
R7(C̄78-82) 47K
R8(Ā72-76) 47K
R9(B̄60-64) 47K
R10(Ā54-58) 47K
R11(F̄48-52) 47K
R12(Ē42-46) 82K
CR16(Ē8-12) P5
CR17(F̄8-12)
CR18(D̄20-24)
CR19(C̄20-24) X01
CR20(B̄20-24) Ff
CR21(Ā78-82) W5
CR22(B̄78-82)
CR23(B̄72-76) X̄01
CR24(C̄72-76)
CR25(Ē78-82) K̄4 K5 K6 W1 W2
CR26(D̄78-82)
CR27(J̄8-12) P0
CR28(H̄8-12)
CR29(B̄60-64) X̄01
CR30(D̄60-64)
CR31(Ē54-58) K̄4 K̄5 K6
CR32(D̄54-58)
CR33(25C-H̄) P̄0
CR34(26C-H̄) Fc
CR35(28C-H̄)
CR36(H̄48-52) X01
CR37(J̄48-52) F̄c
CR38(K̄48-52) Fv
CR39(Ē48-52)
CR40(Ē54-58) P̄5
CR41(F̄54-58)
CR42(C̄42-46) Ēf
CR43(B̄42-46)
CR44(H̄42-46) F
CR45(J̄42-46) Ef
CR46(F̄42-46)
X0
46
+60
R18(A53-57) 47K
R19(A59-63) 47K
R20(E47-51) 47K
R21(C73-77) 47K
R22(E35-39) 47K
R23(D29-33) 82K
CR62(E13-17) P5
CR63(F13-17)
CR64(B53-57) K̄4 K5 K6
CR65(D53-57)
CR66(B59-63) X̄01
CR67(D59-63)
CR68(K53-57) F̄c
CR69(J53-57)
CR70(E47-51) Fc
CR71(H47-51) X01
CR72(K47-51) Fv
CR73(J47-51)
CR74(F53-57) P̄0
CR75(E53-57) P̄5
CR76(H53-57) K4 K5 K6 W1 W2
CR77(E73-77)
CR78(B73-77) S̄5
CR79(A73-77) X01
CR80(D73-77)
CR81(F35-39) X̄01
CR82(H35-39) Ff
CR83(J35-39)
CR84(K41-45) P0
CR85(K35-39)
CR86(K29-33) Ēf
CR87(J29-33)
CR88(E29-33) Ef
CR89(F29-33) L
CR90(H29-33)
X̄0
44
+60
R30(D1-5) 82K
R31(D13-17) 82K
R32(H13-17) 82K
CR119(K7-11) Ā0
CR120(F7-11)
CR121(D7-11)
CR122(B7-11) Fh
CR123(A7-11) M̄ox
CR124(B1-5) W̄5
CR125(C1-5)
CR126(B13-17) W4
CR127(C13-17)
CR128(J13-17)
CR129(K13-17) Ef
m̄ox
+60
R27(Ē2-6) 82K
CR103(H̄2-6) (A)
CR104(J̄2-6) S1
CR105(K̄2-6) A0
mov
+60
R28(E1-5) 82K
CR106(H1-5) (A)
CR107(J1-5) S1
CR108(K1-5) Ā0
m̄ov
P5
SG227 CR130(25B-F) T5
SG227 CR131(26B-F) T4
SG227 CR132(27B-F) T2
SG227 CR133(28B-F) T1
Ef
(Ef)
R48(Ā85-89) 30K
R49(Ā84-88) 30K
-20
(A)
CR141(F20-24) O2
CR142(H̄20-24) O3
CR143(J̄20-24) S5
CR144(K̄20-24) S̄4
K̄4 K̄5 K6 = (tw)
CR166(A29-33) K̄4
CR167(B29-33) K̄5
CR168(C29-33) K6
+60
R36(D̄48-52) 82K
CR148(C̄48-52) (A)
CR149(B̄48-52) S3
CR150(Ā48-52) Ā0
m̄oz
+60
R42(70C-H) 56K
R43(D35-39) 56K
CR159(69C-H) C4
CR160(68C-H) C̄5
CR161(71C-H)
CR162(C35-39) C̄4
CR163(B35-39) C5
CR164(A35-39)
CR165(J73-77) H̄
R44(D̄85-89) 270K
-20
* +35 VWA
2N697 Q3
B C E
Fc
R45(E85-89) 27K
R46(Ē84-88) 27K
-20
+60
R33(J79-83) 18K
CR134(D79-83) Fg
CR135(B79-83) S̄4
CR136(C79-83) S3
CR137(E79-83) S2
CR138(F79-83) S1
CR139(H79-83) H
* +35 VWA
Q1 2N697
B C E
Fm
R34(B84-90) 1.4K 242E
+60
R35(D47-51) 82K
CR145(C47-51) (A)
CR146(B47-51) S3
CR147(A47-51) A0
moz
78

168


NOTES:

1. ALL RESISTORS, UNLESS OTHERWISE NOTED ARE, CB ¼ W 5%.
2. ASSEMBLY DWG. NO. L 536 001 518
3. ALL DIODES, UNLESS OTHERWISI NOTED, ARE HD6621

Figure B-7. Logic, Card No. 7, Schematic Diagram

NOTES:
1. ALL RESISTORS, UNLESS OTHERWISE NOTED, ARE, 1/4 W-5%.
2. ALL DIODES, UNLESS NOTED, ARE HD6621.
3 ASSEMBLY DWG NO. L 536001519.

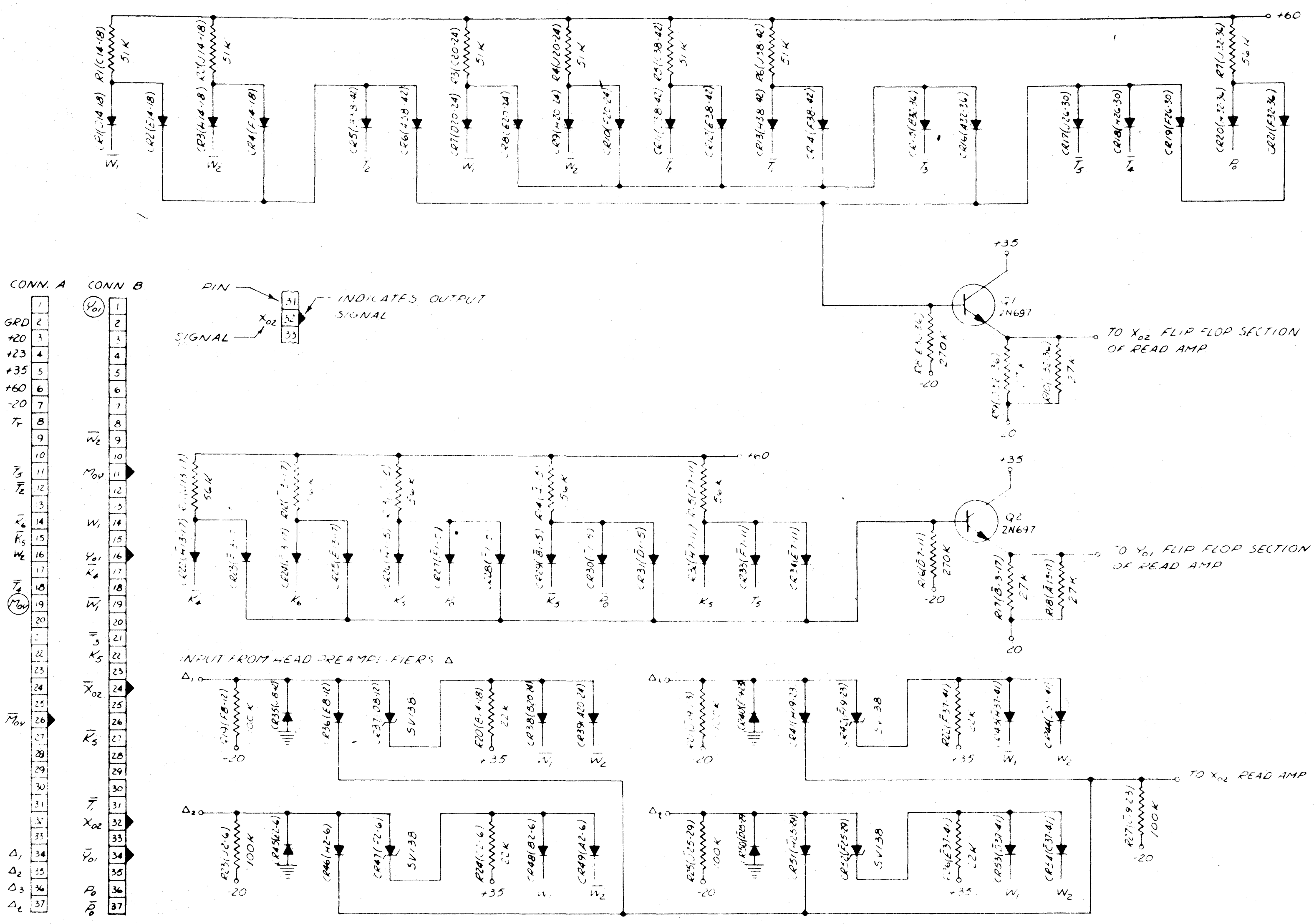


Figure B-8. Logic, Card No. 8, Schematic Diagram

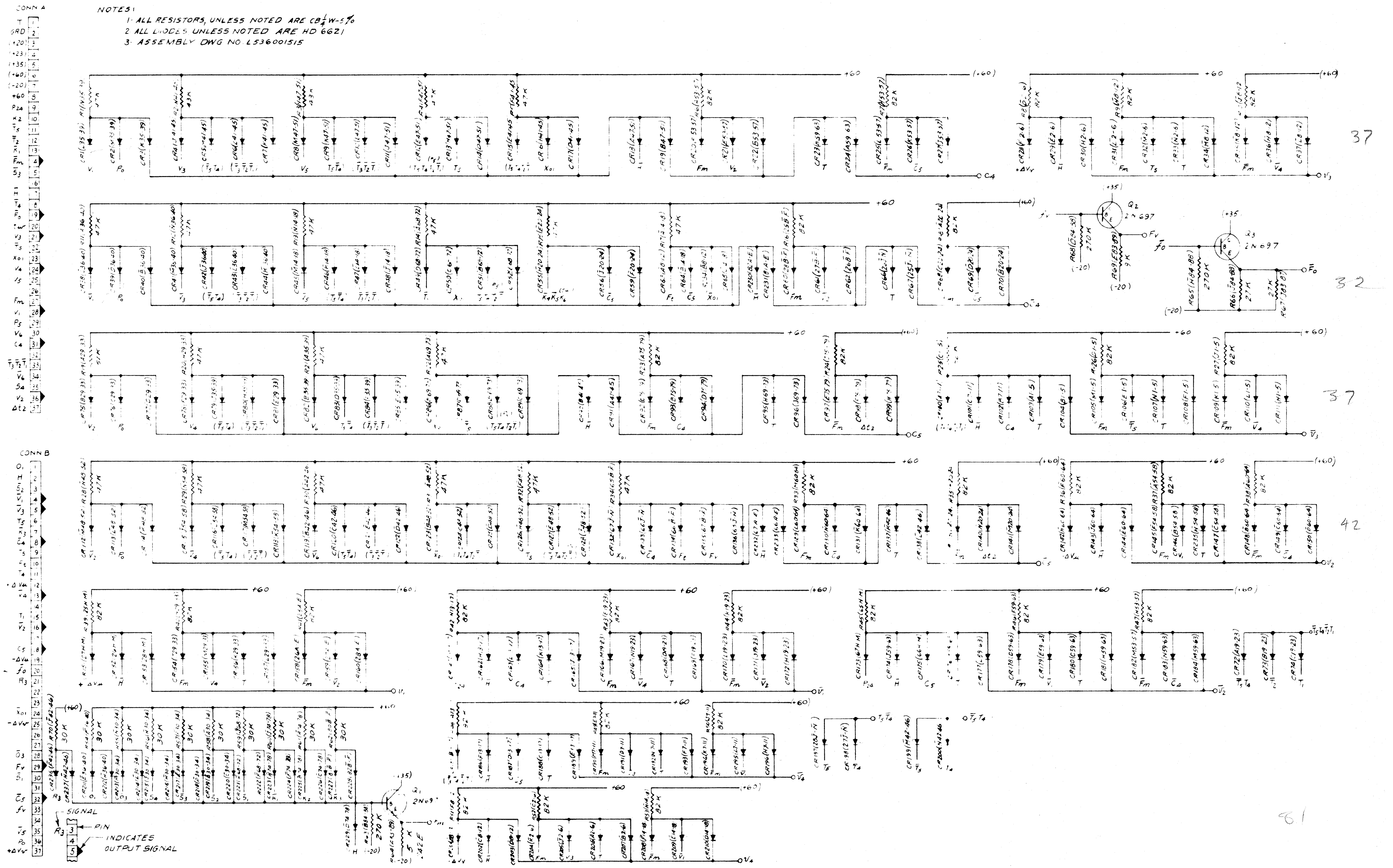


Figure B-9. Logic, Card No. 9, Schematic Diagram

NOTES:

1. ALL RESISTORS ARE ¼W, ±5% UNLESS OTHERWISE SPECIFIED.
2. ALL DIODES, UNLESS NOTED, ARE HD 6621.
3 ASSEMBLY DRAWING NO. L536 001524.

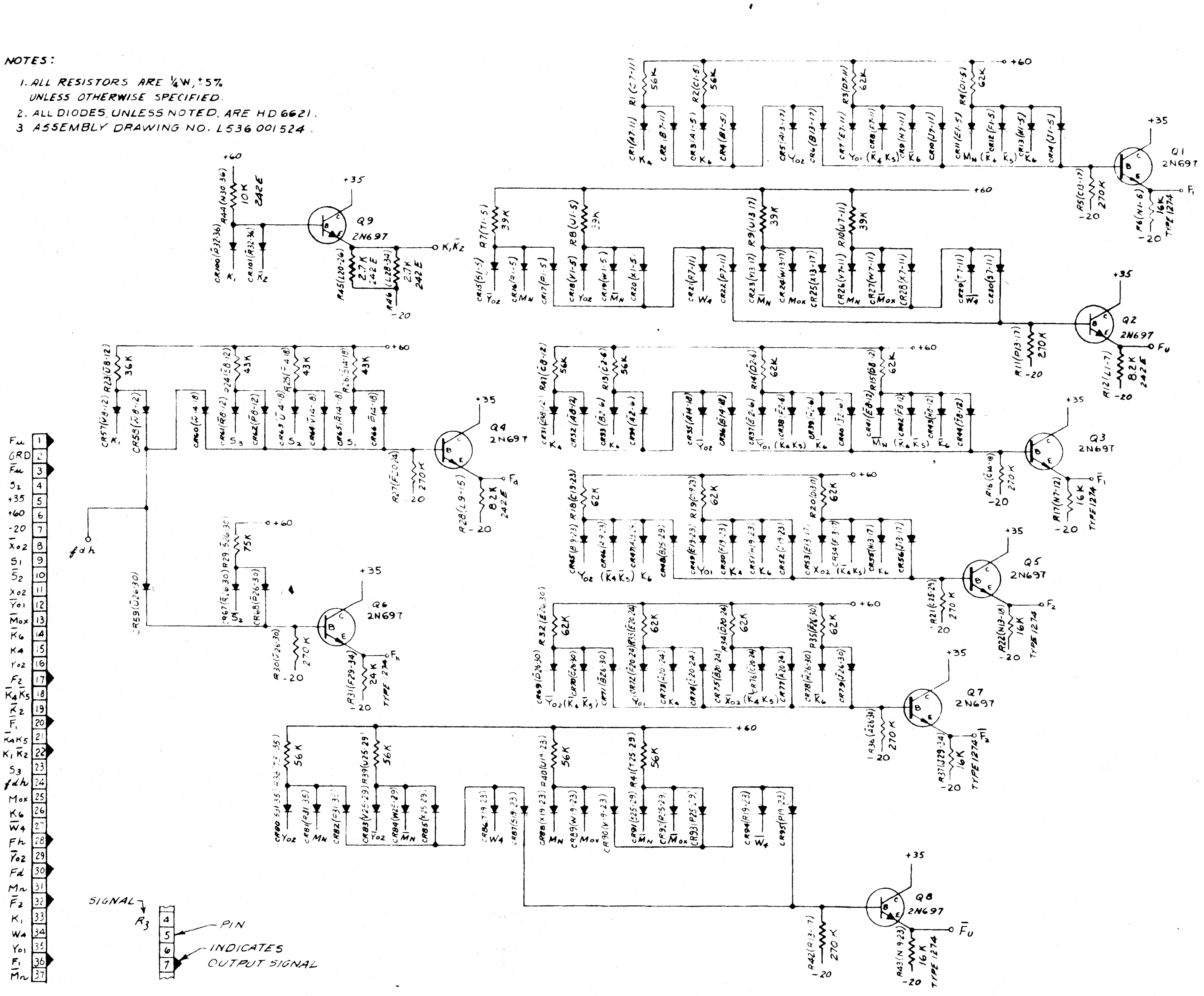
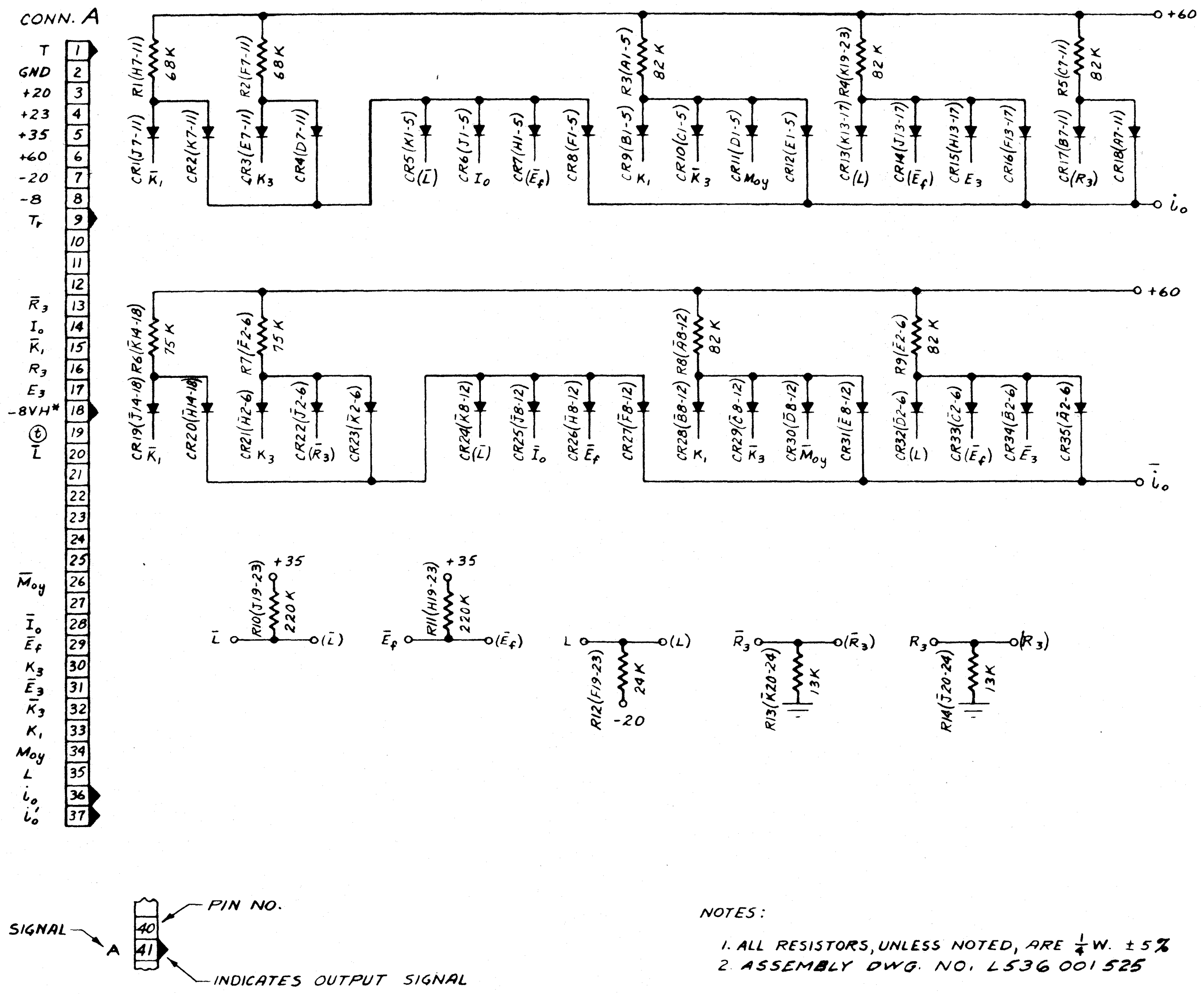


Figure B-10. Logic, Card No. 13, Schematic Diagram

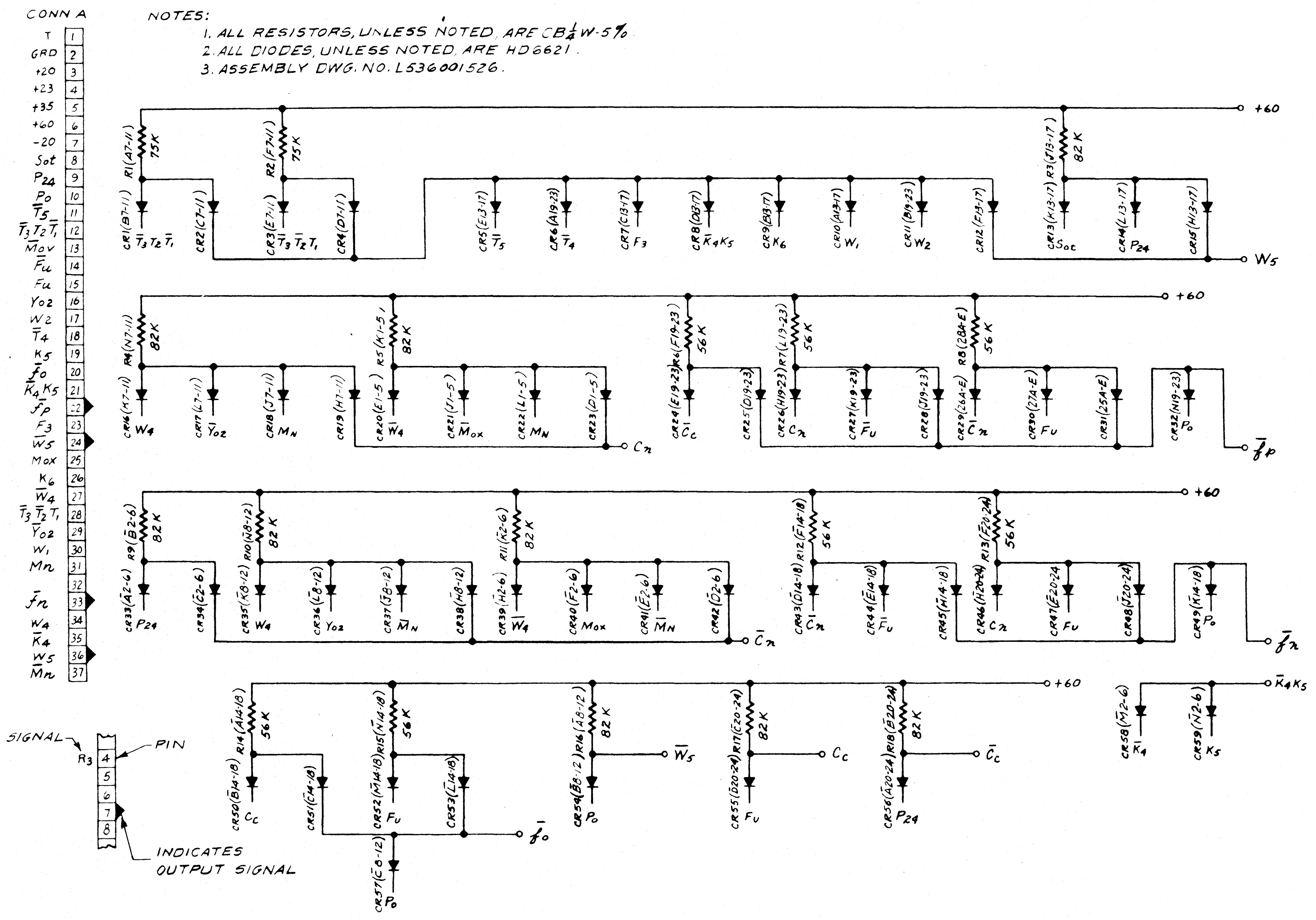
92



NOTES:

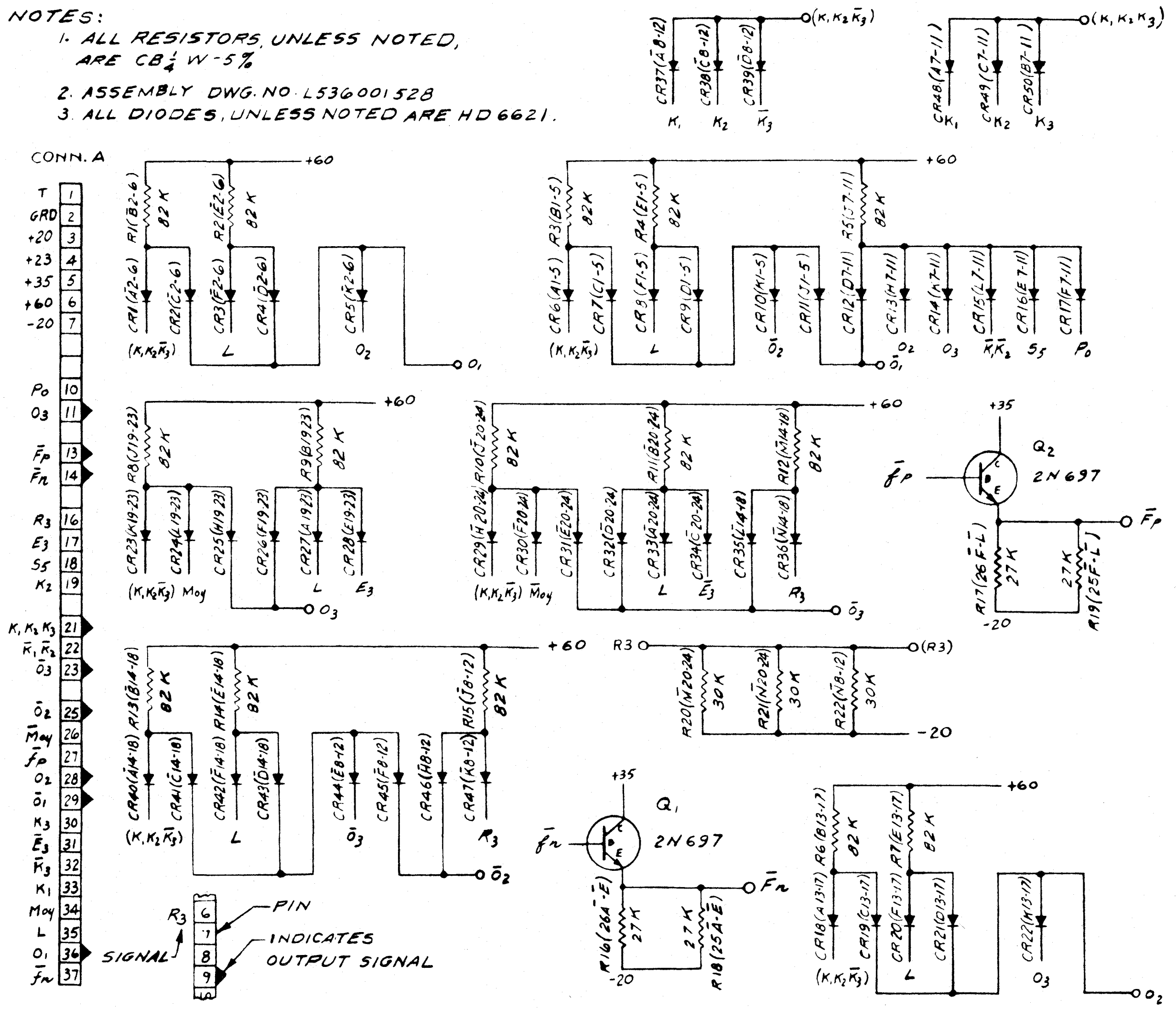
1. ALL RESISTORS, UNLESS NOTED, ARE $\frac{1}{4}$W. ±5%
2. ASSEMBLY DWG. NO. L536 001 525

Figure B-11. Logic, Card No. 14, Schematic Diagram



NOTES:
1. ALL RESISTORS, UNLESS NOTED, ARE CB$\frac{1}{4}$W-5%
2. ALL DIODES, UNLESS NOTED, ARE HD6621.
3. ASSEMBLY DWG. NO. L536001526.

Figure B-12. Logic, Card No. 12, Schematic Diagram



NOTES:
1. ALL RESISTORS, UNLESS NOTED, ARE CB$\frac{1}{4}$ W-5%
2. ASSEMBLY DWG. NO. L536001528
3. ALL DIODES, UNLESS NOTED ARE HD6621.

Figure B-13. Logic, Card No. 16, Schematic Diagram

NOTE:

1. ALL RESISTORS, UNLESS NOTED ARE CB 1/4 W ± 5 %
2. ALL DIODES, UNLESS NOTED ARE HD 6621
3. ASSEMBLY DWG. NO. L536001529.

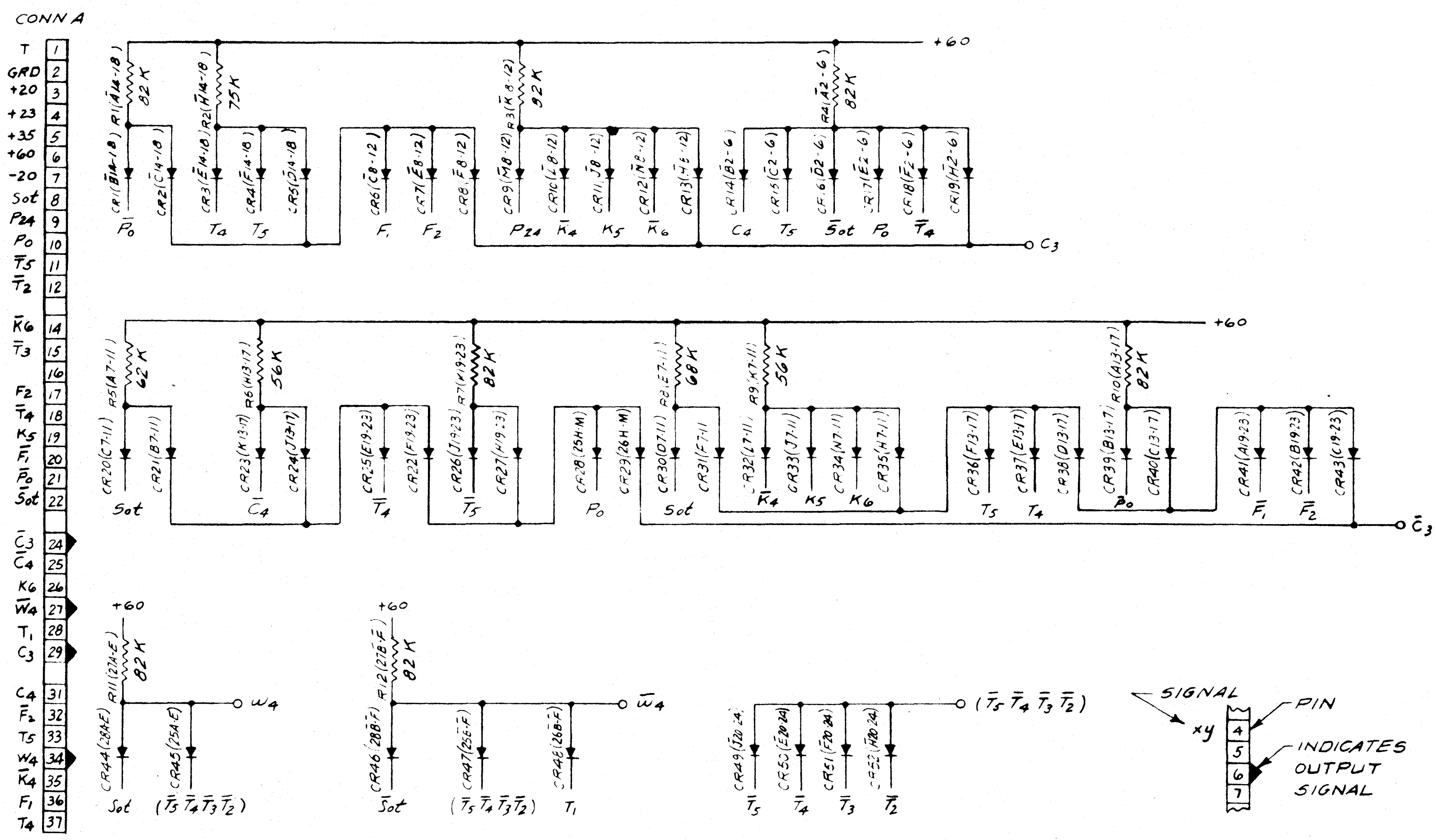


Figure B-14. Logic, Card No. 17, Schematic Diagram

NOTES:

1- ALL RESISTORS, UNLESS NOTED, ARE CB¼W-5%.

2- ALL DIODES UNLESS NOTED, ARE HD 6621.

3- ASSEMBLY DWG. NO. L536001530.

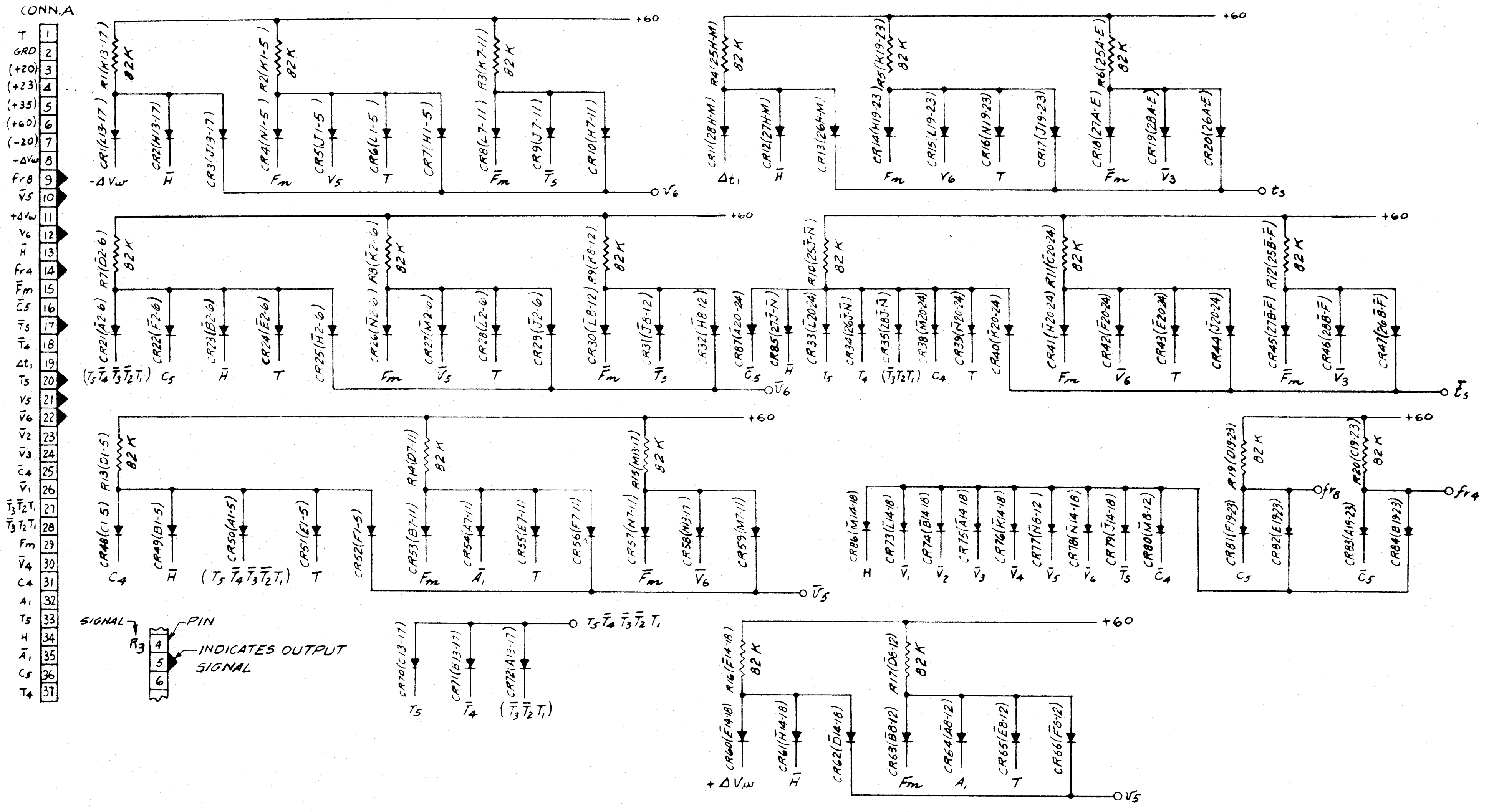
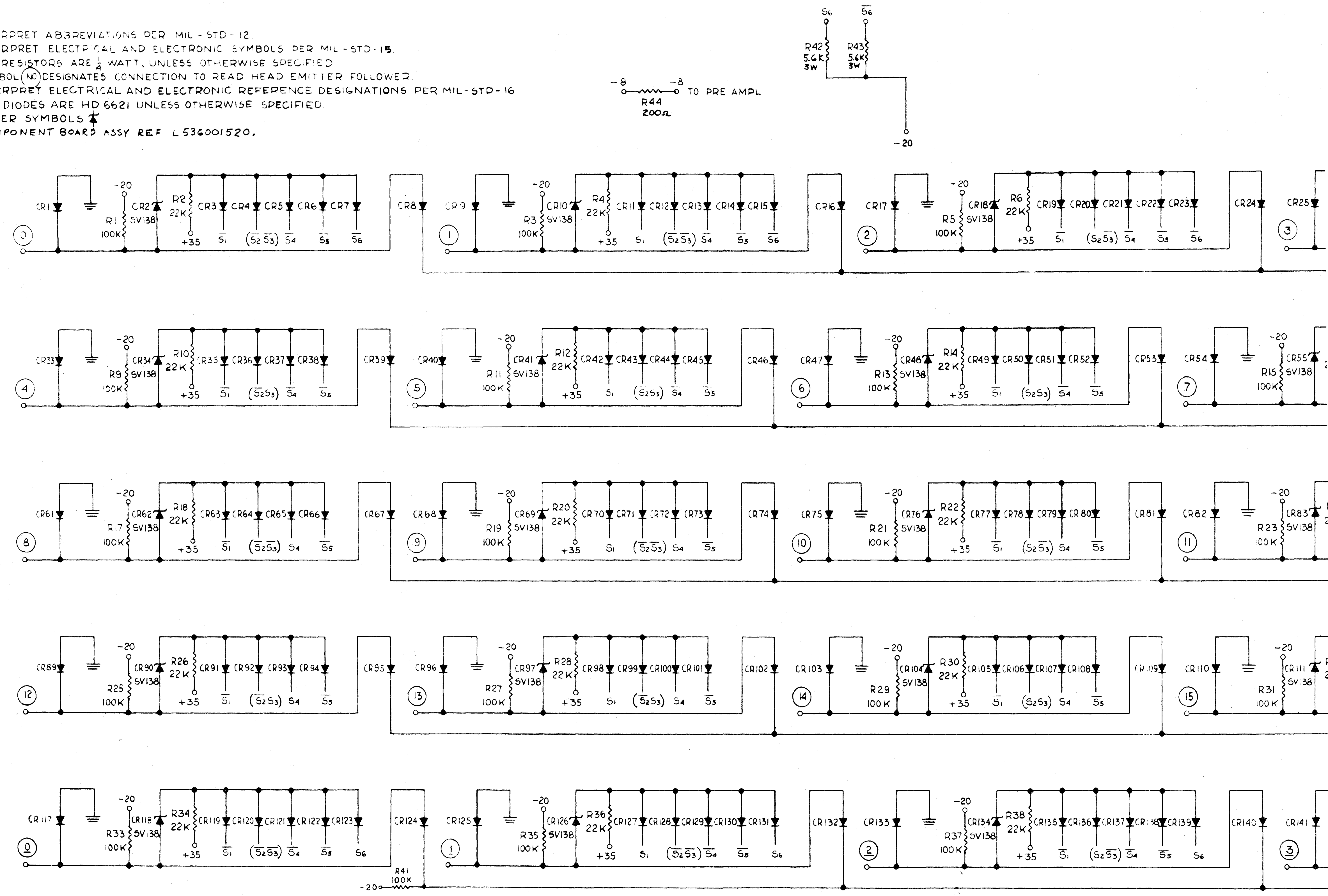


Figure B-15. Logic, Card No. 19, Schematic Diagram

NOTES:

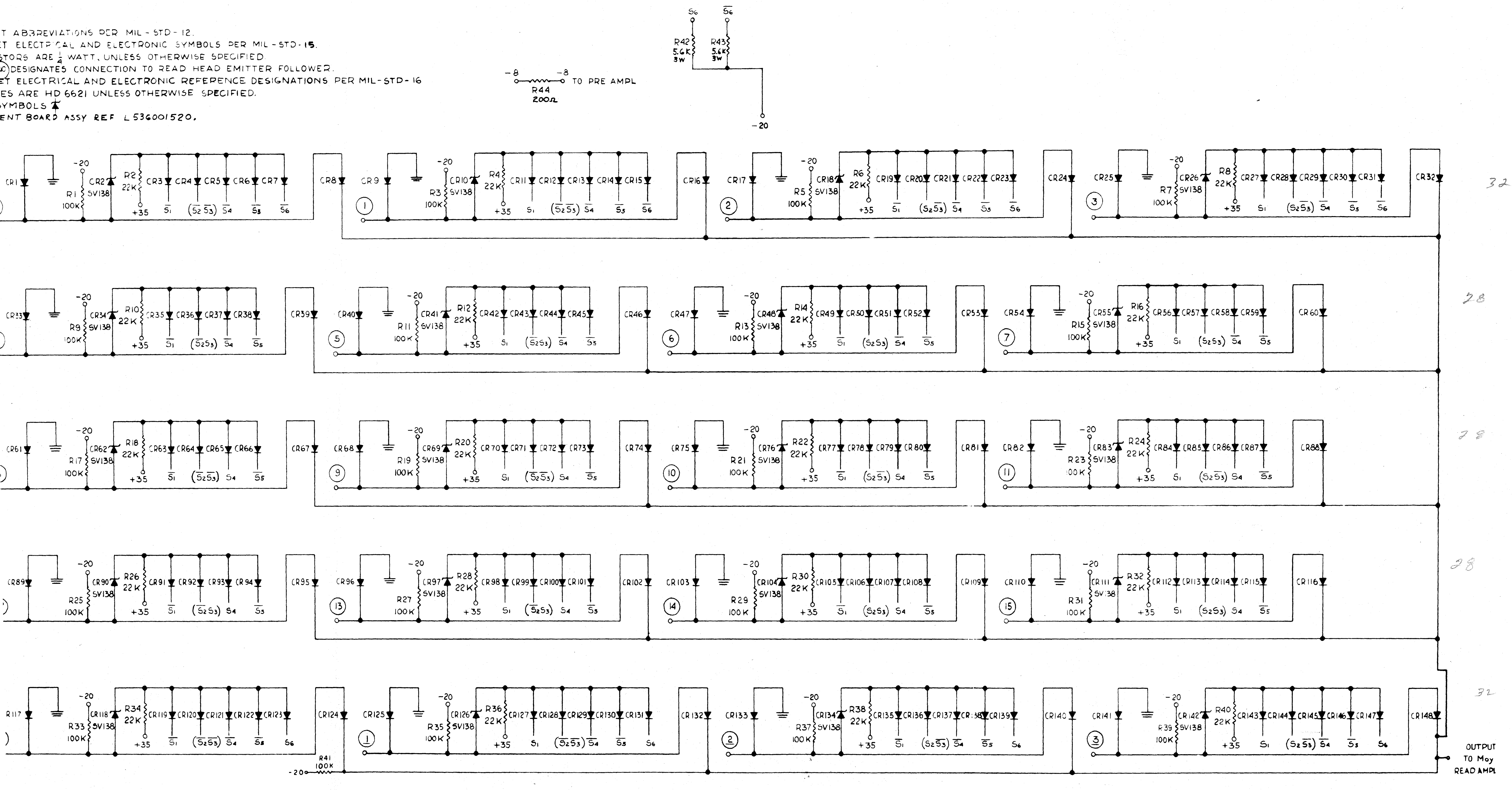
1. INTERPRET ABBREVIATIONS PER MIL-STD-12.
2. INTERPRET ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15.
3. ALL RESISTORS ARE $\frac{1}{4}$ WATT, UNLESS OTHERWISE SPECIFIED
4. SYMBOL (NO) DESIGNATES CONNECTION TO READ HEAD EMITTER FOLLOWER.
5. INTERPRET ELECTRICAL AND ELECTRONIC REFERENCE DESIGNATIONS PER MIL-STD-16
6. ALL DIODES ARE HD 6621 UNLESS OTHERWISE SPECIFIED.
7. ZENER SYMBOLS
8. COMPONENT BOARD ASSY REF L536001520.



GRD	1
+35	2
-20	3
-8	4
$\overline{S}_2S_3$	5
$S_2\overline{S}_3$	6
$\overline{S}_4$	7
S_4	8
-8VH*	9
H15	10
H13	11
H11	12
H9	13
H7	14
H5	15
H3	16
H1	17
H3	18
H1	19
$\overline{S}_5$	20
$\overline{S}_2\overline{S}_3$	21
S_2S_3	22
$\overline{S}_6$	23
S_6	24
$\overline{S}_1$	25
S_1	26
MOY	27
H14	28
H12	29
H10	30
H8	31
H6	32
H4	33
H2	34
Ho	35
H2	36
Ho	37

Figure B-16. Head Selectin

T ABBREVIATIONS PER MIL-STD-12.
T ELECTRICAL AND ELECTRONIC SYMBOLS PER MIL-STD-15.
STORS ARE ¼ WATT, UNLESS OTHERWISE SPECIFIED
)DESIGNATES CONNECTION TO READ HEAD EMITTER FOLLOWER.
ET ELECTRICAL AND ELECTRONIC REFEPENCE DESIGNATIONS PER MIL-STD-16
ES ARE HD 6621 UNLESS OTHERWISE SPECIFIED.
YMBOLS
ENT BOARD ASSY REF L536001520.



S6
S̄6
R42 5.6K 3W
R43 5.6K 3W
-20
-8
R44 200Ω
TO PRE AMPL
CR1 R1 100K -20 CR2 5V138 R2 22K CR3 CR4 CR5 CR6 CR7 +35 S̄1 (S2 S̄3) S̄4 S̄5 S̄6 CR8
CR9 ① R3 100K -20 CR10 5V138 R4 22K CR11 CR12 CR13 CR14 CR15 +35 S1 (S̄2S̄3) S̄4 S̄5 S̄6 CR16
CR17 ② R5 100K -20 CR18 5V138 R6 22K CR19 CR20 CR21 CR22 CR23 +35 S̄1 (S2S̄3) S̄4 S̄5 S̄6 CR24
CR25 ③ R7 100K -20 CR26 5V138 R8 22K CR27 CR28 CR29 CR30 CR31 +35 S1 (S2S̄3) S̄4 S̄5 S̄6 CR32
CR33 R9 100K -20 CR34 5V138 R10 22K CR35 CR36 CR37 CR38 +35 S̄1 (S̄2S3) S̄4 S̄5 CR39
CR40 ⑤ R11 100K -20 CR41 5V138 R12 22K CR42 CR43 CR44 CR45 +35 S1 (S̄2S3) S̄4 S̄5 CR46
CR47 ⑥ R13 100K -20 CR48 5V138 R14 22K CR49 CR50 CR51 CR52 +35 S̄1 (S2S3) S̄4 S̄5 CR53
CR54 ⑦ R15 100K -20 CR55 5V138 R16 22K CR56 CR57 CR58 CR59 +35 S1 (S2S3) S̄4 S̄5 CR60
CR61 R17 100K -20 CR62 5V138 R18 22K CR63 CR64 CR65 CR66 +35 S̄1 (S̄2S̄3) S4 S̄5 CR67
CR68 ⑨ R19 100K -20 CR69 5V138 R20 22K CR70 CR71 CR72 CR73 +35 S1 (S̄2S̄3) S4 S̄5 CR74
CR75 ⑩ R21 100K -20 CR76 5V138 R22 22K CR77 CR78 CR79 CR80 +35 S̄1 (S2S̄3) S4 S̄5 CR81
CR82 ⑪ R23 100K -20 CR83 5V138 R24 22K CR84 CR85 CR86 CR87 +35 S1 (S2S̄3) S4 S̄5 CR88
CR89 R25 100K -20 CR90 5V138 R26 22K CR91 CR92 CR93 CR94 +35 S̄1 (S̄2S3) S4 S̄5 CR95
CR96 ⑬ R27 100K -20 CR97 5V138 R28 22K CR98 CR99 CR100 CR101 +35 S1 (S̄2S3) S4 S̄5 CR102
CR103 ⑭ R29 100K -20 CR104 5V138 R30 22K CR105 CR106 CR107 CR108 +35 S̄1 (S2S3) S4 S̄5 CR109
CR110 ⑮ R31 100K -20 CR111 5V138 R32 22K CR112 CR113 CR114 CR115 +35 S1 (S2S3) S4 S̄5 CR116
R117 R33 100K -20 CR118 5V138 R34 22K CR119 CR120 CR121 CR122 CR123 +35 S̄1 (S̄2S̄3) S̄4 S̄5 S6 CR124
-20 R41 100K
CR125 ① R35 100K -20 CR126 5V138 R36 22K CR127 CR128 CR129 CR130 CR131 +35 S1 (S̄2S̄3) S̄4 S̄5 S6 CR132
CR133 ② R37 100K -20 CR134 5V138 R38 22K CR135 CR136 CR137 CR138 CR139 +35 S̄1 (S2S̄3) S̄4 S̄5 S6 CR140
CR141 ③ R39 100K -20 CR142 5V138 R40 22K CR143 CR144 CR145 CR146 CR147 +35 S1 (S2S̄3) S̄4 S̄5 S6 CR148
OUTPUT TO Moy READ AMPL

32

28

28

28

32

Figure B-16. Head Selecting Matrix, Card No. 10, Schematic Diagram

148